



BIBLIOTECA  
FAC. DE INFORMÁTICA  
U.N.L.P.

**Trabajo de Grado**

# DATA WAREHOUSING

## **Integrantes**

Paula Jaime  
Marcelo Matto  
Ariel Schuster

## **Director**

Prof. Gustavo Rossi

<p><b>TES</b> <b>99/13</b> <b>DIF-02085</b> <b>SALA</b></p>	<div data-bbox="459 1954 525 2040"></div> <div data-bbox="530 1947 872 2056"><p>UNIVERSIDAD NACIONAL DE LA PLATA FACULTAD DE INFORMÁTICA Biblioteca 50 y 120 La Plata catalogo.info.unlp.edu.ar biblioteca@info.unlp.edu.ar</p></div> <div data-bbox="502 2077 750 2161"> DIF-02085</div>
---	---



## Contenido

<b>INTRODUCCIÓN .....</b>	<b>4</b>
Características generales de un Data Warehouse .....	6
Justificación de un Data Warehouse .....	7
Objetivo del trabajo .....	8
 <b>FUNDAMENTOS TEÓRICOS.....</b>	<b>9</b>
EVOLUCIÓN DEL DATA WAREHOUSING.....	10
1.1 Historia del análisis de datos .....	10
1.1.2 Los datos desde los sistemas Legales .....	10
1.1.3 Información procesada en PC's .....	11
1.1.4 Sistemas de Soporte de Decisión y de Información Ejecutiva.....	11
1.2 Factores de evolución del Data Warehousing.....	12
1.3 Cambios en la naturaleza del negocio .....	13
ATRIBUTOS Y CONCEPTOS .....	14
2.1 El entorno arquitectónico .....	14
2.2 Estructurando datos en el data warehouse .....	17
2.3 El ciclo de vida del desarrollo.....	20
2.4 El fenómeno día 1 a N.....	21
2.5 Granularidad.....	22
2.6 Particionamiento .....	22
EL DATA WAREHOUSE Y SU DISEÑO.....	25
3.1 Migración de datos operacionales.....	25
3.2 El Data warehouse y los modelos de datos .....	28
3.3 El modelo de datos.....	30
3.4 El modelo de datos de nivel medio .....	31
3.5 El modelo de datos físico.....	34
3.6 Normalización y desnormalización.....	34
3.7 Snapshots en el data warehouse.....	35
3.8 Metadatos (Metadata).....	36
3.9 Manejo de tablas de referencia en el data warehouse.....	37
3.10 Ciclicidad de datos .....	37
3.11 Registros perfiles .....	38
3.12 Modelo Dimensional: Esquemas estrella.....	39
LA TECNOLOGÍA Y EL DATA WAREHOUSE.....	49
4.1 Tipos DBMS y el Data Warehouse .....	52
4.2 Bases de datos Multidimensionales y el Data Warehouse .....	53



<b>METODOLOGÍA.....</b>	<b>54</b>
<b>METODOLOGÍA DE DISEÑO .....</b>	<b>55</b>
Paso 1: Análisis del modelo de datos.....	55
Paso 2: Análisis Breadbox .....	56
Paso 3: Distribución Técnica.....	56
Paso 4: Preparación del entorno técnico.....	57
Paso 5: Análisis del área objetivo.....	57
Paso 6: Diseño del data warehouse.....	58
Paso 7: Análisis del sistema fuente.....	58
Paso 8: Especificación del programa .....	59
Paso 9: Programación.....	60
Paso 10: Población .....	60
 <b>UN EJEMPLO DE DISEÑO UTILIZANDO LA METODOLOGÍA.....</b>	 <b>61</b>
<b>DEFINICIÓN DEL DOMINIO .....</b>	<b>62</b>
6.1 Definición del S.I.A. ....	62
6.2 Funcionalidad del S.I.A.....	62
6.3 Circuito Administrativo General.....	64
6.4 Descripción funcional de los módulos.....	65
6.5 Tablas de clasificación de datos.....	72
 <b>DESARROLLO DE LA METODOLOGÍA .....</b>	 <b>81</b>
7.1 Pasos del desarrollo.....	81
Paso 1: Análisis del modelo de datos.....	81
Paso 2: Análisis breadbox.....	83
Paso 3: Distribución Técnica.....	83
Paso 4: Preparación del entorno Técnico.....	83
Paso 5: Análisis del área objetivo.....	85
Paso 6, 7: Diseño del Data Warehouse – Análisis del sistema fuente.....	86
Paso 8-Especificaciones.....	90
Paso 9-Programación.....	91
Paso 10-Población .....	93
 <b>CONCLUSIONES .....</b>	 <b>95</b>
Trabajo a futuro.....	98
 <b>ANEXO .....</b>	 <b>102</b>
 <b>REFERENCIAS.....</b>	 <b>115</b>



---

## Introducción

---

^ Durante varios años el acceso a la información corporativa para el análisis y posterior toma de decisiones por parte del personal calificado para dichas tareas, fue un punto débil de la informática, ya que los sistemas existentes para tal objetivo no permitían realizar este tipo de tareas eficientemente o ni si quiera se podían realizar. Debido a estas necesidades es que surge la idea de un “*Almacén de datos corporativo*” o “*Data Warehouse*” con una estructura distinta a la Base de Datos operacional, adecuada para la explotación de la información de Gestión. ^

( En su nivel más elemental un **Data Warehouse** (Almacén de Datos) es un área de transferencia de información para soporte de decisiones. Un **Data Warehouse** recopila datos provenientes de diversas aplicaciones en los sistemas corporativos en actividad; integra los datos en un modelo lógico de áreas comerciales temáticas; almacena la información de modo tal que sea accesible y comprensible para los tomadores de “*decisiones no técnicos*” , provee la información a los tomadores de decisiones en todo el ámbito corporativo a través de distintas herramientas de consulta y generación de informes. ^

Una buena decisión siempre requiere de acceso a buena información. Pero determinar la mejor forma de facilitarla puede ser muy difícil. Muchos de los problemas de las empresas se deben a que los usuarios – los mismos gerentes y empleados – no pueden acceder a la información que necesitan, o cuando la consiguen es tarde o de dudosa confiabilidad. Por eso surge el **data warehousing**, proceso por el cual los datos de una empresa se transforman en información estratégica, a la que además se puede acceder en forma sencilla en el momento que se necesite. ^

( Debemos aclarar que el Data Warehouse es un proceso no un producto, define un nuevo concepto para el almacenamiento de datos, integra toda la información generada en una organización, facilitando de esta manera la toma de **decisiones estratégicas**. )



( La información que se genera en una empresa proviene de diferentes fuentes, formatos y tipos, que se consolidan y se transforman. Estos datos deben ser extraídos y transportados hacia el data warehouse, limpios y sincronizados, de manera que desde él queden a disposición de los usuarios. \

Los sistemas operacionales son segmentados en aplicaciones proveyendo solo vistas parciales de las entidades del negocio ( como cliente, producto ). En un banco, por ejemplo, la información concerniente a los ahorros, créditos y depósitos de un cliente son almacenados en aplicaciones separadas. En contraste, un data warehouse provee una vista universal del cliente. El usuario final tiene acceso a la información integrada y analítica con la que puede tomar mejores decisiones e implementar acciones de marketing correctas.

/ Un data warehouse es una herramienta competitiva que da al usuario final la habilidad de acceder a los datos de toda la empresa, provee una representación integrada de las múltiples fuentes de información de la empresa. Esto asegura la consistencia de las reglas de administración y convenciones aplicadas a los datos. Un data warehouse es, por lo tanto, un reflejo de las necesidades corporativas más que las individuales. Luego habrá que proveer a los usuarios de las Herramientas de Análisis. Reporting, Data Sets para **Datamining** o **Datamarts** adecuados a sus necesidades para poder acceder a la información a través del data warehouse \

/ En un data warehouse, un dato es variante en el tiempo. Los usuarios tienen acceso no sólo a la información corriente que está siendo constantemente actualizada en sistemas OLTP, sino también a información histórica. Los datos también son protegidos, ya que se accede a información de sólo lectura. \

Las aplicaciones de soporte de decisión son más fáciles de manejar en un data warehouse que en sistemas legales. Extraer datos desde las aplicaciones de producción construidas en diferentes momentos sobre diferentes bases de datos con codificación diferente es dificultoso y consume mucho tiempo. En el clima de negocios de hoy en día, con el crecimiento de fusiones y adquisiciones de compañías, las aplicaciones de producción heterogéneas tienen que convivir todas juntas. (El data warehouse resuelve el problema de inconsistencia de datos capturándolos en un punto central de almacenamiento y creando un elemento consistente y lógico para distribución de datos operacionales. )

El usuario del data warehouse es el analista DSS cuyo trabajo primario es la definición y descubrimiento de información usada en la toma de decisiones corporativas.

## Características generales de un Data Warehouse

Podemos mencionar algunas características generales de los Data warehouses:

- **Los datos son organizados por área temática** éstas representan una recopilación de todos los datos en la organización que pertenecen a un tema que puede ser de importancia para el tomador de decisiones.
- **Los datos son integrados**, a fin de ser útiles, los datos deben ser transformados a formatos comunes de medida, referencia y almacenamiento, generando así su integración.
- **El Warehouse no es volátil**, la información se carga en el warehouse, y posteriormente se accede para la toma de decisiones. Esto difiere de un sistema transaccional dado que este es actualizado, tan pronto como ocurre un nuevo evento.
- **La información está orientada al tiempo** el warehouse es una secuencia de snapshots (fotografías instantáneas) de información corporativa tomadas a intervalos predefinidos - todos los días, una vez por semana, una vez por mes, etc. -.

El data warehouse es muy usado en las áreas de Marketing y Comercial (Clientes, Productos, Campañas y Acciones Promocionales) y también en Control de Gestión (Canales de Distribución, Análisis por Centro de Costo, Producto y Cliente). También se aplica en áreas tales como Gestión del Riesgo, Morosidad, Reporting, o Gestión Financiera.

La forma más común de analizar la información es utilizando una herramienta de proceso de análisis en línea conocida como On-line Analytical Procesing (**OLAP**). Los productos OLAP ofrecen un rango muy variado de capacidades de análisis avanzado, como el multidimensional y el estadístico.

Un data warehouse soporta también sofisticadas operaciones de análisis que se conocen con el término de **DataMining**.

Una de las novedades que aporta el data warehousing como sistema de análisis de información es la creación del Meta Información (**metadata**). Se trata de un archivo que es considerado como diccionario de estructuras de datos que el administrador del sistema define con el objetivo de asistir en los procesos de consulta a la base de datos. La metadata se adaptará a las definiciones que el usuario utilizará posteriormente en sus consultas al sistema. De esta manera, conseguirá minimizar los complejos procedimientos de definición de nombres de campos, jerarquías y relaciones.

El proceso de implementación de un sistema data warehouse puede adaptarse de forma gradual o departamental. De esta manera se crean soluciones específicas para cada área con el objetivo de conseguir resultados operativos a corto plazo. Esta solución departamental se conoce con el nombre de DataMart.

( Un data warehouse tiene que estar preparado para **crecer permanentemente** para poder absorber los cambios en datos, fuentes o nuevos requerimientos de los usuarios. Además debe ser **escalable** debido al crecimiento continuo de la información, de fácil acceso. Se deben poder agregar rápidamente más dimensiones en DataMart (Rolap/Molap), ejecutar procesos de DataMining sin inconvenientes en los DataSets adecuados, también asegurar la calidad del dato y no interferir los procesos transaccionales. )

## Justificación de un Data Warehouse

( Las razones fundamentales por las cuales conviene crear un data warehouse son de dos tipos:

- De **negocio**: el data warehouse es una importante herramienta para enfrentar los nuevos escenarios caracterizados por la globalización y la competencia, una mayor cultura y exigencia del cliente, aumento en la calidad de servicio, y rapidez y calidad en la toma de decisiones.
- De **tecnología**: el data warehouse brinda a los sistemas de una empresa abaratamiento, madurez y nueva generación de herramientas.

El futuro del data warehouse está muy ligado al desarrollo tecnológico. Un centro de información no alcanzará su objetivo real hasta que no se haga extensible a la totalidad de los centros de decisión de la empresa. Esto será posible única y exclusivamente mediante la utilización de tecnologías Internet/Intranet (Web, Navegadores, etc.).

Las tecnologías de análisis se enriquecerán con usos cada vez más sofisticados de las tecnologías de DataMining y su ampliación cada vez mayor y más dispar de datos.

( Hay que tener en cuenta que un data warehouse es una herramienta eficaz para la organización y análisis de grandes volúmenes de información que las empresas generan, y que posteriormente les permite al desarrollo de estrategias más efectivas y rentables. )

## Objetivo del trabajo

El principal objetivo del trabajo es fundamentar la necesidad de contar con un ambiente de Data Warehouse para el análisis de información estratégica inaccesible mediante los sistemas transaccionales actuales (OLTP). Con el propósito de alcanzar dicho objetivo:

- Haremos una breve reseña de la evolución de los los sistemas de información desde sus orígenes hasta la actualidad mostrando que el crecimiento del volumen de la información a través de los años por sí solo no cumple con los requerimientos de calidad de información para la toma de decisiones en el mercado competitivo actual, ya que para dicha toma de decisiones se necesita que la información sea cada vez de mayor calidad;
- Desarrollaremos conceptos teóricos fundamentales para encarar el desarrollo de un Data Warehouse: arquitectura, modelo, migración de datos, etc.;
- Describiremos la metodología de diseño adoptada basada en el diseño de base de datos multidimensional. La metodología está conformada por una serie de actividades o pasos que mantienen un cierto orden, en donde existe siempre una actividad precedente y otra siguiente, un tiempo estimado para su realización y una frecuencia de ejecución
- Partiendo de un modelo Relacional Real correspondiente a un sistema orientado al manejo de obras sociales (S.I.A), proponemos el diseño de un Data Warehouse utilizando la metodología antes mencionada, para la futura implementación de un módulo de gestión del mismo que en la actualidad no posee, mostrando como se beneficiaría a los tomadores de decisiones con el uso del mismo.

---

## **Sección 1**

### **Fundamentos Teóricos**

---

Esta sección está compuesta de cuatro capítulos. En el Capítulo 1, Evolución del Data Warehousing, veremos la administración histórica del análisis de los datos y los factores que han guiado la evolución del Data Warehousing.

En el Capítulo 2, Atributos y Conceptos, presentaremos los atributos y conceptos claves en el desarrollo de un Data Warehouse, como su ciclo de vida, granularidad, particionamiento, etc.

En el Capítulo 3, El Data Warehouse y su Diseño, veremos que hay dos aspectos fundamentales en la construcción del Data warehouse- el diseño de la interfaz proveniente de los sistemas operacionales y el diseño del Data warehouse en sí mismo.

Por último, en el Capítulo 4, La Tecnología y el Data Warehouse, explicaremos los diferentes tipos de motores de bases de datos utilizadas en la implementación de un Data Warehouse.

## Capítulo 1

# EVOLUCIÓN DEL DATA WAREHOUSING

En este capítulo veremos la administración histórica del análisis de los datos y los factores que han guiado la evolución del Data Warehousing.

## 1.1 Historia del análisis de datos

En la evaluación de esta evolución, debemos comenzar con una revisión de qué se ha hecho con los datos antes de que surgieran los Data Warehouses.

A través de la historia del desarrollo de sistemas, el énfasis ha estado en los sistemas operacionales y en los datos que éstos procesan. No es práctico mantener datos indefinidamente en estos sistemas, y sólo como idea tardía fue una estructura diseñada para archivar los datos que el operacional ha procesado. Los requerimientos fundamentales de los sistemas operacionales y de análisis son diferentes: los sistemas operacionales necesitan performance, mientras que los sistemas de análisis necesitan flexibilidad y gran alcance. Raramente ha sido aceptable que el análisis del negocio interfiera y degrade la performance de los sistemas operacionales.

### 1.1.2 Los datos desde los sistemas Legales

En los 70's todos los desarrollos de sistemas fueron hechos sobre mainframes IBM usando herramientas tales como Cobol, CISCs, IMS, DB2.etc.. Los 80's trajeron consigo las nuevas mini-computadoras tales como AS/400 y VAX/VMS. A finales de los 80's y principio del '90 UNIX se convirtió en una plataforma muy popular con la introducción de la arquitectura cliente/servidor.

No obstante todos los cambios en las plataformas, arquitecturas, herramientas y tecnologías, una gran cantidad de aplicaciones de negocio continúan ejecutándose en

los mainframes de los '70, hay muchas razones para esto, la más importante y relevante en nuestro caso, es que, con los años, estos sistemas han crecido capturando el conocimiento y reglas del negocio que es increíblemente dificultoso de migrar a una nueva plataforma o aplicación.

Estos sistemas, genéricamente llamados sistemas legales, son la más grande fuente de datos para sistemas de análisis. Los datos que están almacenados en DB2, IMS, VSAM, etc. para los sistemas transaccionales terminan en grandes bibliotecas de cintas en centros de datos remotos. Una institución generará reportes y extractos durante años, cada uno diseñado para extraer información que no está en los sistemas locales. En muchos casos, grupos IS/IT asumen la responsabilidad de diseñar y desarrollarlos programas para estos reportes. El tiempo requerido para generar estos programas frecuentemente es mucho mayor de lo que el usuario final piensa que puede llevar.

### **1.1.3 Información procesada en PC's**

Durante la década pasada, la popularidad creciente de las PC's ha introducido muchas opciones y oportunidades para el análisis del negocio. Los usuarios más avanzados comenzaron a usar en sus PC's programas de base de datos que les permitía almacenar y trabajar con información extraída desde los sistemas legales. Muchas herramientas de reportes y análisis fueron dirigidas a usuarios finales entre los que obtuvieron gran popularidad.

El lado negativo de este modelo para el análisis del negocio es que deja los datos fragmentados y orientados a necesidades muy específicas. Cada usuario individual ha obtenido sólo la información que él/ella requería, al no estar estandarizadas, las extracciones no son capaces de satisfacer los requerimientos de múltiples usuarios.

### **1.1.4 Sistemas de Soporte de Decisión y de Información Ejecutiva**

Los Sistemas de Soporte de Decisión se han enfocado más sobre los detalles y están orientados a managers de jerarquía media y baja. Los Sistemas de Información Ejecutiva han provisto un mayor nivel de consolidación y una vista multi-dimensional de los datos, ya que los ejecutivos de mayor jerarquía necesitan analizar los datos haciendo *slice and dice* más que revisar los detalles.

Estas dos categorías son quizás las más cercanas precursoras de los sistemas de data warehousing. Las siguientes son algunas características asociadas con los sistemas DSS y EIS:

- Estos sistemas tienen datos descriptivos en lugar de nombres de atributos crípticos (utilizados en los sistemas tradicionales). Los nombres y estructuras de datos están diseñados para ser usados por personas no técnicas.
- Los datos son generalmente preprocesados con la aplicación de reglas de negocio estándar.
- Ofrecen vistas consolidadas de datos, tales como productos, clientes, etc. Aunque estos sistemas tienen la habilidad de acceder al detalle de los datos (drill down) raramente permiten acceder a todos los datos detallados al mismo tiempo.

Los sistemas de data warehousing actuales proveen las herramientas analíticas proporcionadas por estos precursores, pero su diseño no es derivado de analistas o ejecutivos y, como veremos mas adelante, los sistemas de data warehousing tienen más éxito cuando satisfacen la estructura de negocio en conjunto más que requerimientos específicos.

## 1.2 Factores de evolución del Data Warehousing

Muchos factores han influido en la rápida evolución de la disciplina de data warehousing. El conjunto de factores más importante ha sido el avance tecnológico en hardware y software. La disminución de los precios y el potencial del hardware, junto con la facilidad de uso del software actual, han hecho posible el rápido análisis de cientos de gigabytes de información.

Resumiendo los factores más importantes en la evolución del data warehousing son:

- Los procesadores de las computadoras de hoy son más poderosos y más baratos que los mainframes de ayer.
- Los dispositivos de almacenamiento pueden almacenar cientos gigabytes de información a bajo costos.
- Las PC's de escritorio son más poderosas para soportar herramientas de análisis.
- Los precios de hardware y software han bajado abruptamente. )



### **1.3 Cambios en la naturaleza del negocio**

( Otra influencia significativa en la evolución del data warehousing fueron los cambios fundamentales en la organización y estructura de los negocios en los últimos diez años. El surgimiento de una economía global ha cambiado las demandas de información hechas por las corporaciones del mundo entero. Estas han encontrado mercado para sus productos globalmente compitiendo con otras compañías de diferentes culturas y entornos económicos.

La baja económica de finales de los '80 condujo a muchas corporaciones globales a un remarcable período de consolidación. Fenómenos tales como “proceso de reingeniería de negocio” o “downsizing” condujeron a las compañías a reevaluar sus prácticas de negocio para identificar áreas de competencia y evitar negocios que no fueran productivos. La globalización de los negocios ha incrementado la necesidad no solamente de análisis más continuo, sino manejar datos en forma centralizada. )

## Capítulo 2

# ATRIBUTOS Y CONCEPTOS

Después de haber visto el uso histórico de los datos de análisis y explorado los factores que influenciaron la evolución del data warehousing, en este capítulo presentaremos los atributos y conceptos claves de un data warehouse.

## 2.1 El entorno arquitectónico

En el núcleo del entorno arquitectónico del data warehouse hay fundamentalmente dos tipos de datos: primitivos y derivados, las principales diferencias entre ellos son las siguientes:

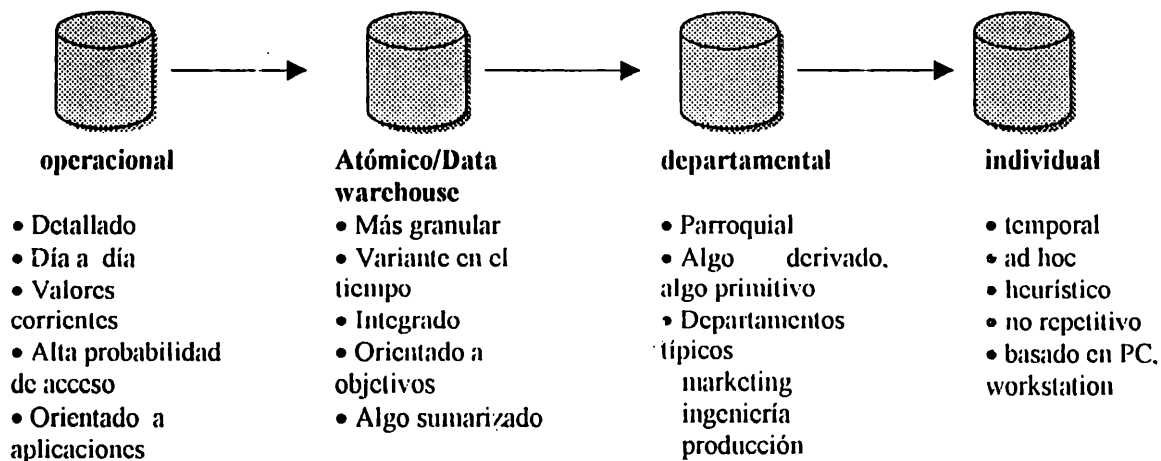
### Datos Primitivos / Datos Operacionales

- Orientados a la aplicación
- Detallados
- Precisos al momento de acceso
- Sirven a los usuarios comunes
- Pueden ser actualizados
- Ejecutan repetitivamente
- Los requerimientos para procesamiento son entendidos a priori
- Compatibles con el SDLC
- Sensitivos a la performance
- Accedidos una unidad a la vez
- Manejados por transacciones
- Alta disponibilidad
- Manejados en su totalidad
- No redundantes
- Estructura estática, contenido variable
- Pequeñas cantidades de datos usadas en un proceso
- Soporta operaciones día a día
- Alta probabilidad de acceso

### Datos Derivados/Datos DSS

- Orientados a objetivos
- Sumarizados, refinados
- representan valores en el tiempo, snapshots
- sirven a los directivos
- no son actualizados
- ejecutan heurísticamente
- los requerimientos para procesamiento no son entendidos a priori
- Ciclo de vida completamente diferente
- Performance relajada
- Accedidos un conjunto a la vez
- Conducido por análisis
- Disponibilidad relajada
- Manejados por subconjuntos
- La redundancia es un hecho de vida
- Estructura flexible
- Grandes cantidades de datos usadas en un proceso
- Soporta necesidades gerenciales
- Baja probabilidad de acceso

La extensión natural de la división de datos causada por la diferencia entre datos primitivos y derivados es mostrada en la siguiente figura:



Hay cuatro niveles en la arquitectura: operacional, atómico o data warehouse, departamental e individual. El nivel operacional mantiene datos primitivos solamente y sirve al procesamiento transaccional de alta performance. El data warehouse mantiene datos primitivos que son actualizados y algunos datos derivados. El nivel departamental de datos contiene datos derivados casi exclusivamente, y en el nivel individual es donde se hace mucho análisis heurístico.

El entorno departamental, a veces llamado datamart, OLAP o DBMS multidimensional, contiene información útil a diferentes departamentos de la compañía. Hay una base de datos departamental de marketing, contable, auditoría, etc. El origen de todos los datos departamentales es el data warehouse. Un aspecto importante del entorno arquitectónico es la integración que ocurre a través de la arquitectura. )

El data warehouse es la base del procesamiento DSS. El trabajo del analista DSS en el entorno del data warehouse es mucho más fácil que en el entorno clásico porque hay una fuente de datos simple e integrada y porque los datos son accesibles. Un data warehouse es una colección de datos para el soporte de toma de decisiones que es orientada a objetivos, integrada, no volátil y variante en el tiempo.

( El data warehouse está orientado a las principales áreas objetivo de la corporación que han sido definidos en el modelo de datos. Los más comunes son: cliente, producto, actividad, póliza, reclamo, cuenta. )

El área objetivo termina siendo físicamente implementada como una serie de tablas relacionadas en el data warehouse que pueden residir en diferentes medios. Los datos que tienen una alta probabilidad de acceso y un bajo volumen de

almacenamiento residen en un medio que es rápido y relativamente caro; los datos que tienen una baja probabilidad de acceso y de gran volumen residen en un medio que es barato y de acceso más lento. El DASD y la cinta magnética son los medios más populares, aunque también están los discos ópticos y las fichas (para almacenar datos que nunca han sido reproducidos en un medio electrónico).

(Cuando los datos son organizados en base a un objetivo cada clave tiene un elemento de tiempo. Algunas tablas son organizadas sobre una base fecha-a-fecha, llamada organización continua de datos; otras tablas son organizadas sobre una base acumulativa mensual y otras sobre una fecha individual, pero todos los registros tienen alguna forma de fecha vinculada a la clave.)

El aspecto más importante del data warehouse es que *es integrado*, esta integración se realiza cuando los datos pasan del entorno operacional al data warehouse.)

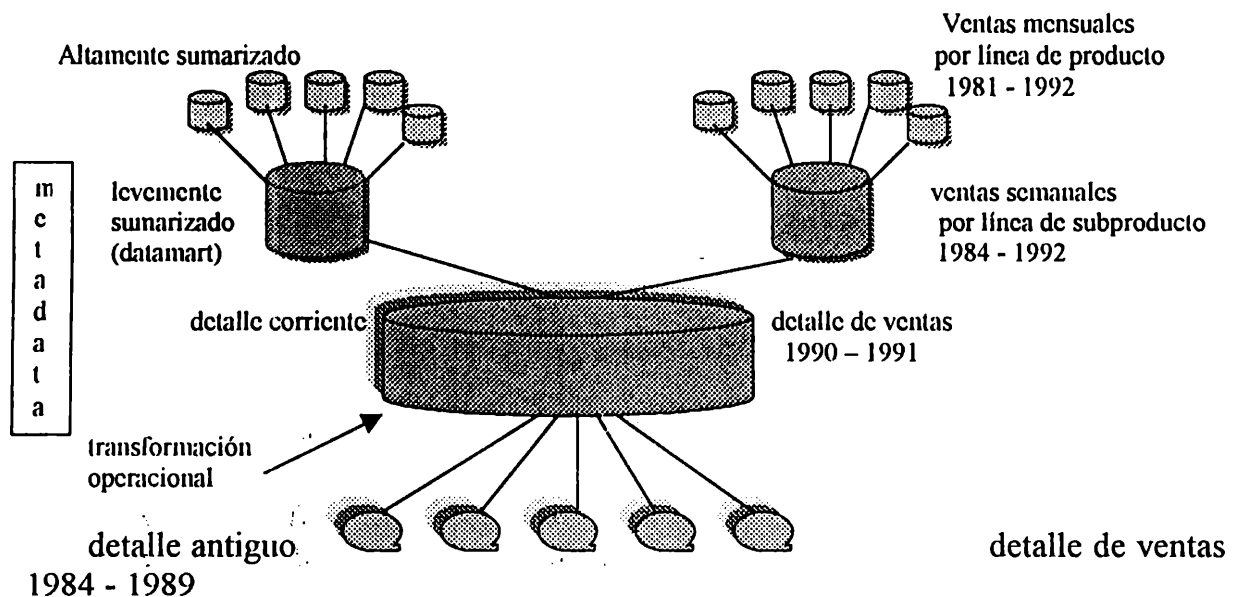
(Otra característica es que *es no volátil*, el data warehouse es cargado generalmente en masa para consulta, no se realizan actualizaciones.)

(La última característica importante del data warehouse es que *es variante en el tiempo*). El horizonte de tiempo para el data warehouse es significativamente mayor que para los sistemas operacionales. Un horizonte de 30 - 90 días es normal en sistemas operacionales, mientras que para el data warehouse es normal un horizonte de 5 a 10 años.

(Las bases de datos operacionales contienen datos con “valores corrientes”, cuya exactitud es válida al momento de acceso con lo cual pueden ser actualizados. Los datos del data warehouse son una serie de snapshots tomados en un momento en el tiempo.)

(La estructura de clave de los datos operacionales pueden o no contener elementos de tiempo (año, mes, día). En el data warehouse siempre contiene un elemento de tiempo.)

El data warehouse se estructura como muestra la figura:



Hay diferentes niveles de detalle en el data warehouse: un nivel de detalle más antiguo (usualmente almacenado en cinta), un nivel corriente, uno levemente sumariado (el datamart) y uno altamente sumariado. Los datos fluyen desde el entorno operacional al data warehouse sufriendo un significativa transformación.

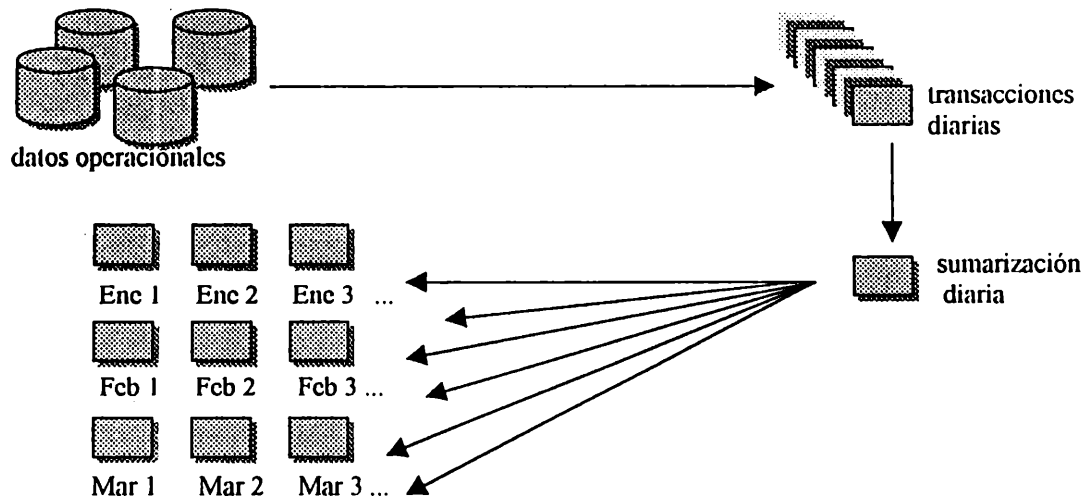
A medida que pasa el tiempo los datos pasan del detalle corriente al detalle más antiguo; y cuando son sumariados pasan del detalle corriente al levemente sumariado y luego al altamente sumariado.

## 2.2 Estructurando datos en el data warehouse

Hay muchas formas de estructurar los datos dentro del data warehouse:

- Acumulativa simple
- Rolling summary
- Directa simple
- Continua

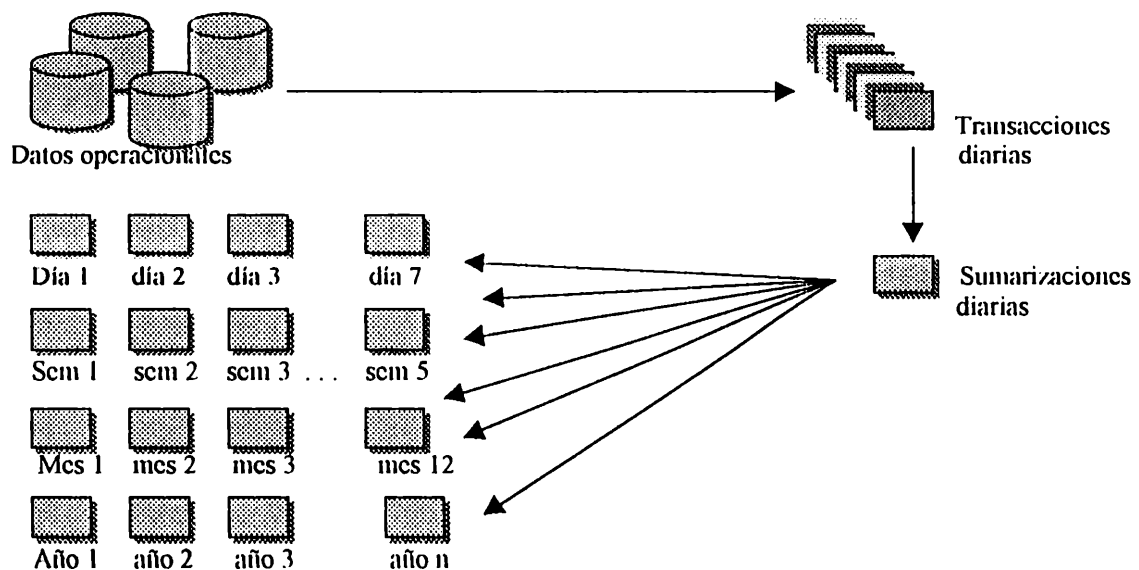
La más común es la estructura acumulativa simple, como muestra la siguiente figura, en la que, diariamente, las transacciones son transportadas desde el entorno operacional y sumariadas en registros del data warehouse. La sumariación puede ser por cliente, cuenta, u otra área objetivo y por fecha. En otras palabras, toda la actividad diaria para un cliente para una cuenta es totalizada y pasada al data warehouse en una base día por día.



Una variación es la llamada *storage of rolling summary data*, en donde los datos pasan desde el operacional como antes, pero en los primeros siete días de la semana la actividad es sumariada en siete slots diarios, el octavo día los siete slots diarios son sumados y ubicados en el primer slot semanal y los totales diarios son sumados en el primer slot diario. Al final de mes, los slots semanales son sumados y ubicados en el primer slot mensual, luego los slots semanales son reseteados a cero. Al final del año, los slots mensuales son sumados y el primer slot anual es cargado y los slots mensuales son reseteados a cero.

Una estructura rolling summary data maneja mucho menos unidades de datos que la estructura acumulativa simple.

La siguiente figura muestra este tipo de estructuración de datos:



Otra posibilidad para la estructuración de datos en el data warehouse es el archivo directo simple, en donde los datos son pasados desde el operacional sin acumulación y sin una base diaria, es hecho en un largo período de tiempo, como una semana o un mes, es decir, representa un snapshot de datos operacionales tomados en un instante de tiempo.

De dos o más archivos simples puede ser creado un archivo continuo en donde están los datos desde el primero al último mes.

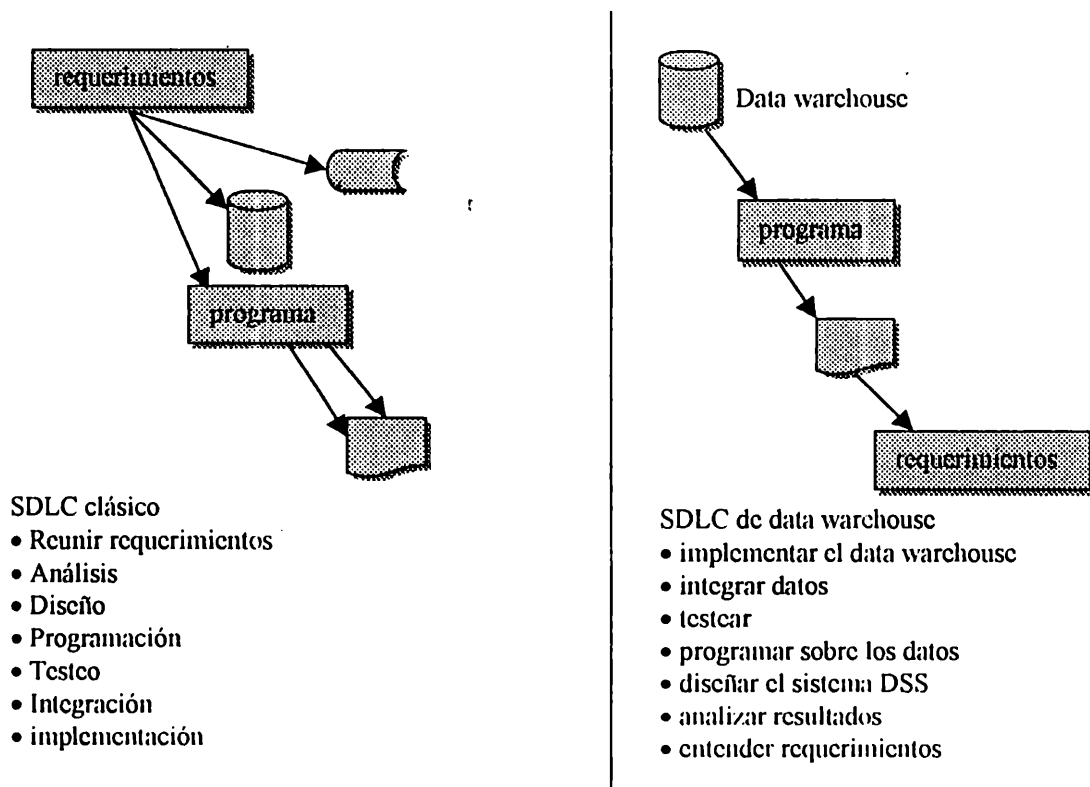
Las claves del data warehouse son inevitablemente compuestas, porque casi siempre la fecha es parte de la clave y, porque los datos son particionados, los distintos componentes de las particiones son parte de la clave.

## 2.3 El ciclo de vida del desarrollo

El ciclo de vida de desarrollo de sistemas clásico (SDLC) no funciona en el mundo del analista DSS. El SDLC asume que los requerimientos son conocidos al comienzo del diseño (o al menos pueden ser descubiertos). Sin embargo, en el mundo del analista DSS los requerimientos usualmente es la última cosa descubierta en el ciclo de vida de desarrollo; hay un ciclo de vida muy diferente asociado al data warehouse.

Mientras los datos operacionales son usualmente no integrados y en el data warehouse deben ser integrados, hay otras diferencias entre el nivel de datos y procesamiento operacional y el nivel de datos y procesamiento del data warehouse.

Una de ellas es con respecto a los ciclos de desarrollo como muestra la figura:





El entorno operacional es soportado por el ciclo de desarrollo clásico (SDLC) mientras que el data warehouse opera de forma diferente, a veces llamado CLDS (a la inversa).

El SDLC clásico es conducido por requerimientos, primero se deben entender los requerimientos, luego se diseña y se desarrolla. El CLDS es a la inversa, se comienza con los datos a los que se integran y testean, se escriben los programas para ellos, los resultados de éstos son analizados y, finalmente, los requerimientos del sistema son entendidos. El CDLS es un ciclo de desarrollo conducido por datos.

## **2.4 El fenómeno día 1 a N**

El data warehouse no es construido todo a la vez, es diseñado y poblado de a un paso a la vez, es evolucionante. El proceso típico de construir un data warehouse es el siguiente:

- El día uno hay un políglota de sistemas haciendo esencialmente procesamiento operacional.
- El día dos, las primeras tablas de la primera área objetivo del data warehouse son pobladas, en este punto se despierta la curiosidad y los usuarios comienzan a descubrir el data warehouse y el procesamiento analítico.
- El día tres, más tablas del data warehouse son pobladas y con ello surgen más usuarios que encuentran una fuente de datos integrada que es fácil de obtener y tiene una base histórica hecha mirando los datos en un espectro de tiempo.
- El día cuatro, algunos de los datos que residieron en el entorno operacional son ubicados apropiadamente en el data warehouse, y éste es ahora “descubierto” como una fuente para hacer procesamiento analítico. La competencia para acceder al data warehouse comienza a ser un obstáculo.
- El día cinco, comienzan a surgir las bases de datos departamentales (datamart u OLAP).
- El día seis, como es más fácil, rápido y barato obtener datos departamentales que desde el data warehouse, los usuarios pasan del detalle del data warehouse al procesamiento departamental.

- El día n, la arquitectura está totalmente desarrollada. Hay pocos usuarios directos del data warehouse y muchas bases de datos departamentales en donde se realiza el procesamiento analítico DSS.

Esta evolución toma varios años.

## 2.5 Granularidad

El aspecto más importante del diseño de un data warehouse es la granularidad, que se refiere al nivel de detalle o sumariazación mantenido en el data warehouse, a mayor detalle menor nivel de granularidad y a menor detalle mayor nivel de granularidad.

La razón por la cual la granularidad es el principal tema de diseño en el entorno del data warehouse es que afecta al volumen de datos que reside en el data warehouse y al tipo de consultas que pueden ser respondidas.

Muchas veces se necesita eficiencia en el almacenamiento y acceso a los datos y analizar datos en gran detalle. Cuando una organización tiene grandes cantidades de datos, tiene sentido considerar dos niveles de granularidad en la porción detallada del data warehouse: datos levemente detallados y datos detallados de archivo. En el nivel de datos detallados de archivo, se almacena todo el detalle que viene del entorno operacional, por esta razón se almacenan en cinta.

Por los costos, eficiencia, fácil acceso y habilidad para responder consultas, el nivel dual de datos es la mejor elección arquitectónica para el nivel detallado del data warehouse para muchos negocios. Solamente cuando un negocio tiene una cantidad de datos relativamente pequeña en el data warehouse puede intentarse un nivel de datos simple.

## 2.6 Particionamiento

(Un segundo tema de diseño de datos en el data warehouse es el particionamiento, que se refiere a la división de datos en unidades físicas separadas que pueden ser manejadas independientemente.

Se dice que si la granularidad y particionamiento son hechos apropiadamente, casi todos los aspectos del diseño e implementación del data warehouse son fáciles. )

En el entorno del data warehouse, la cuestión no es si el detalle corriente será particionado sino cómo será particionado. Cuando los datos residen en grandes unidades físicas, entre otras cosas no pueden ser:

- Reestructurados fácilmente
- Indexados libremente
- Secuencialmente examinados, si es necesario
- Reorganizados fácilmente
- Recuperados fácilmente
- Monitoreados fácilmente

En resumen, una de las esencias del data warehouse es el acceso flexible a los datos, teniendo una gran cantidad de datos disminuye este propósito, por lo tanto todos los datos del detalle corriente serán particionados.

Hay muchos criterios por los que los datos pueden ser divididos:

- Por fecha
- Por línea de negocios
- Por geografía
- Por unidad organizacional
- Por todo lo de arriba

Esta elección la hace el desarrollador, pero (es casi obligatorio particionar por fecha.) También debe decidir si particionar a nivel de sistema o aplicación. El particionamiento a nivel de sistema es una función del DBMS y el sistema operativo, el particionamiento a nivel de aplicación es hecho por el código y estrictamente controlado por el desarrollador y programador.

Como regla, tiene sentido particionar datos a nivel de aplicación, la razón más importante es que a nivel de aplicación puede haber una definición diferente cada año que puede o no ser la misma. La naturaleza de datos en un warehouse es la colección de datos en un largo período de datos.

Cuando el particionamiento es hecho a nivel de sistema, el DBMS inevitablemente requiere una definición de datos simple. Dado que el data warehouse mantiene datos por un largo período de tiempo – hasta 10 años – y dado que la definición regularmente cambia, no tiene sentido permitir que el DBMS o el sistema operativo tengan una definición de datos simple.

Otra característica importante al permitir el particionamiento da nivel de aplicación es que los datos pueden ser movidos desde un complejo procesamiento a otro, esto

es una ventaja cuando la carga de trabajo y el volumen de datos llegan a ser un problema.

El test para particionar datos es responder a la pregunta: puede ser agregado un índice a una partición sin interrumpir otras operaciones? Si no puede ser agregado, entonces necesita ser particionada más finamente.



## Capítulo 3

# EL DATA WAREHOUSE Y SU DISEÑO

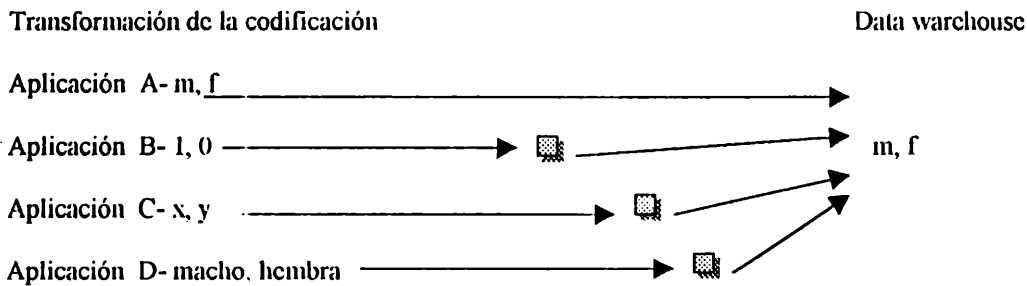
Hay dos aspectos fundamentales en la construcción del Data warehouse- el diseño de la interfaz proveniente de los sistemas operacionales y el diseño del Data warehouse en sí mismo. En principio, se carga con una porción de datos, luego el analista del sistema de soporte para la toma de decisiones (DSS) lo usa y analiza, luego los datos se modifican u otros datos se agregan al Data warehouse, estas modificaciones o agregados se basan en el feedback ( retroalimentación ) con el usuario final, este feedback es constante durante toda la vida del Data warehouse.

## 3.1 Migración de datos operacionales

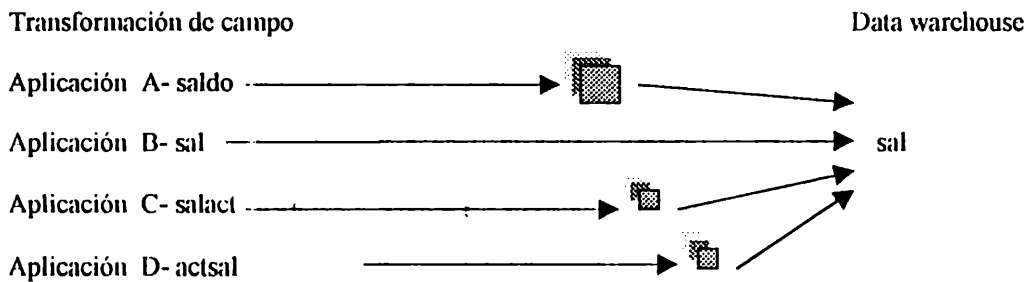
Es bastante (común pensar\ que el proceso de migración desde el ambiente operacional al Data warehouse consiste en extraer los datos operacionales e ingresarlos al warehouse, este pensamiento está bastante alejado de la realidad\ya que existen varias razones por las que no se puede realizar el traslado de esta forma.

La primer dificultad con las que nos encontramos es que los datos en el ambiente operacional no están integrados, cuando se construyeron las aplicaciones existentes no se pensó en su futura posible integración. Cada aplicación tenía su propio conjunto de requerimientos únicos y privados y no se tuvieron en cuenta otras aplicaciones durante el desarrollo, en consecuencia nos encontramos con que algunos datos existen en varios lugares con diferentes nombres, algunos datos se llaman igual pero son diferentes en distintos lugares, algunos datos se llaman igual y significan lo mismo pero están codificados distintos. Hay una gran cantidad de detalles que se deben programar para traer los datos desde el ambiente operacional en forma correcta. Podemos mencionar como ejemplo simple de la falta de integración datos que no están codificados en forma consistente, como la codificación del genero. En una codificación el genero se codifica como "m/f", en otra se codifica como "0/1". Realmente no importa como se codifica el genero en el Data warehouse siempre y cuando sea consistente. (Cuando se pasa al Data

warehouse, las distintas codificaciones se deben descifrar y recodificar con el valor correcto). En la figura se muestra la transformación de la codificación



Otro tema involucrado en la integración es la transformación de campos. El mismo campo existe en cuatro aplicaciones con cuatro nombres distintos. A continuación vemos dicha transformación:



Estos dos ejemplos son muy sencillos en sí mismos pero multiplicados por los miles de archivos y sistemas existentes el tema de la integración se convierte en pesado y complejo de manejar. \

Pero el tema de la integración no es la única dificultad en la transformación de los datos desde el ambiente operacional, hay otro problema relacionado con la eficiencia en el acceso a los datos de los sistemas existentes). La pregunta que surge es ¿Cómo sabe el programa que busca en los sistemas existentes si ya se ha buscado en un determinado archivo?. Existe mucha información en el ambiente operacional como para estar buscando en todos los archivos cada vez que se tiene que realizar una búsqueda del data warehouse.

Hay tres tipos de cargas que se deben realizar al data warehouse desde el ambiente operacional:

- la carga de datos históricos.
- la carga de datos contenidos actualmente en el ambiente operacional.
- la carga de cambios al ambiente del data warehouse de cambios (actualizaciones) que han ocurrido en el ambiente operacional desde el último refresco del data warehouse. \

La carga de datos históricos presenta un desafío mínimo porque no se realiza a menudo. La carga de datos del ambiente operacional existente también presenta un desafío mínimo porque se necesita hacer sólo una vez. La carga de datos que se debe realizar cuando se producen cambios en el ambiente operacional presenta el mayor desafío para el “*arquitecto de datos*”. La búsqueda de archivos existentes es un tema fundamental que tiene que enfrentar el arquitecto de los datos.

(Hay cinco técnicas comunes usadas para limitar la cantidad de datos examinados: \

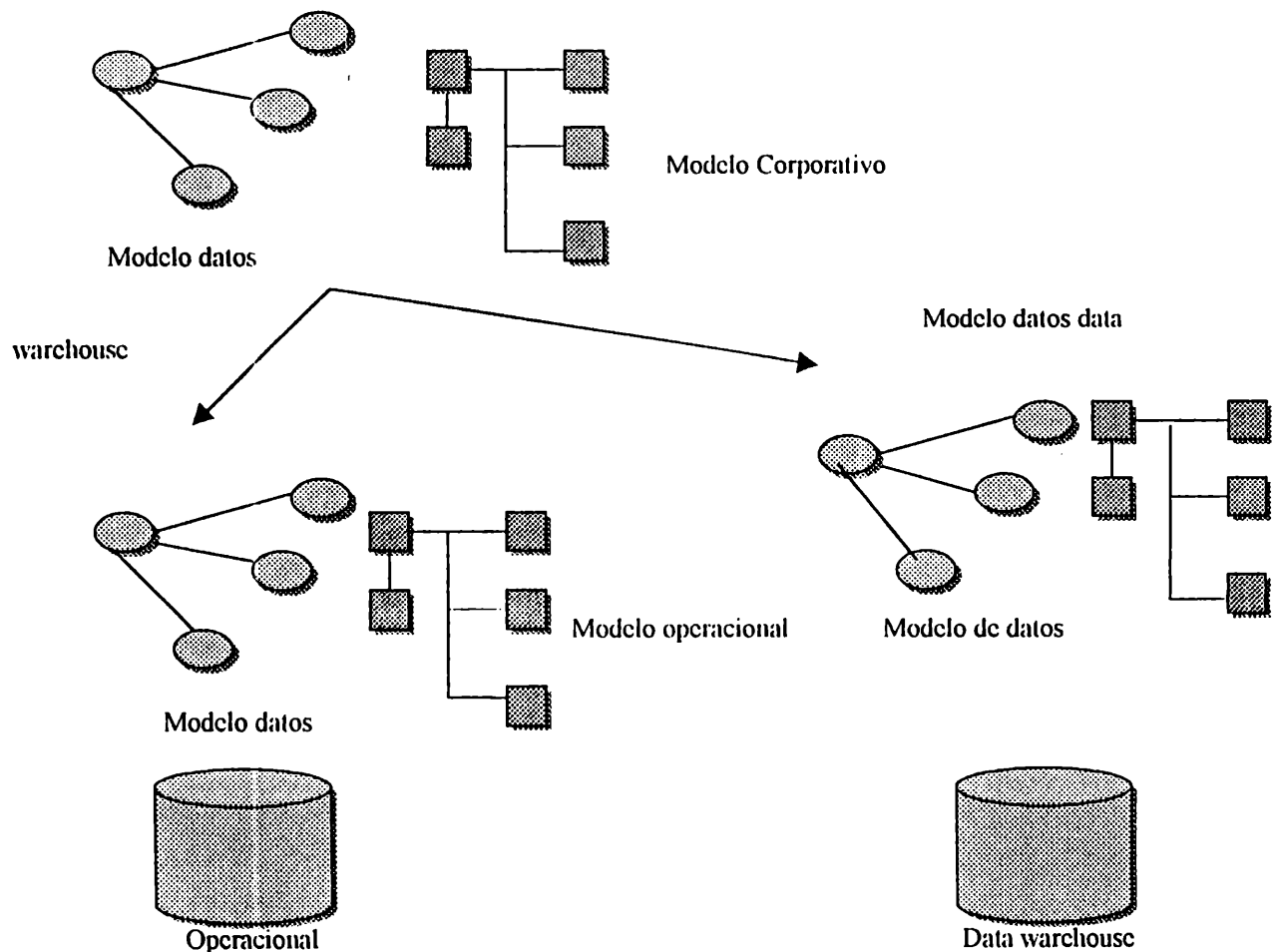
- (La primera técnica es buscar los datos a los que se les haya colocado una marca de la hora.) Cuando una aplicación coloca, estampa la hora del último cambio o actualización en un registro, la búsqueda del data warehouse puede realizarse bastante eficientemente porque los datos que tengan una hora que no se corresponda con la que se está aplicando no deben tocarse. Sin embargo, generalmente no es común que se les coloque la hora a los datos existentes.
- La segunda técnica para limitar la cantidad de datos a examinar para la extracción del data warehouse es buscar en un archivo “*delta*”. Dicho archivo lo crea una aplicación y contiene sólo los cambios realizados a una aplicación. Cuando hay un archivo “*delta*” el proceso de búsqueda es muy eficiente porque los datos que no son candidatos a ser examinados nunca se tocan. El problema es que no muchas aplicaciones construyen archivos “*delta*”.
- La tercera técnica es la de buscar en un archivo de auditoría o log. El archivo de auditoría o log contiene, en esencia, la misma información que un archivo delta, sin embargo hay diferencias importantes. Muchas veces, las operaciones protegen a los archivos de log porque se necesitan en el proceso de recuperación. Otra dificultad es que el formato interno está construido con propósitos de sistemas y no de aplicaciones. Otra desventaja con los archivos de log es que, generalmente, contienen mucha más información que la que desea el desarrollador del data warehouse. Los archivos de auditoría tienen muchas de las mismas desventajas que los archivos de log.
- (La cuarta técnica es modificar el código de la aplicación, aunque no es una opción muy aceptada ya que el código es frágil y viejo.
- (La quinta y última opción es la de tener un archivo imagen de “*antes*” y “*después*”). En esta opción se toma una fotografía ( snapshot ) en el momento de la extracción. Cuando llega el momento de realizar otra extracción se toma otra fotografía y se comparan serialmente ambas fotografías para determinar si hubo cambios o no, pero esta alternativa requiere de una gran cantidad de recursos además de ser pesada y compleja.

(Mencionamos la integración y la performance como dos temas que hacen difícil el proceso de extracción del ambiente operacional, otra dificultad importante es el desplazamiento en base al tiempo. Los datos operacionales son casi siempre de valor actual, esto significa que los datos son válidos al momento de acceder a ellos y pueden ser actualizados pero los datos que ingresan al data warehouse no pueden ser actualizados por lo tanto esta información debe tener un elemento del tiempo asignado a ella.)

### 3.2 El Data warehouse y los modelos de datos

(El modelo de datos es aplicable tanto al ambiente operacional como al data warehouse.)

En la siguiente figura podemos ver que hay un modelo de datos corporativo global que se construyo sin tener en cuenta la diferenciación entre los sistemas operacionales y el data warehouse.

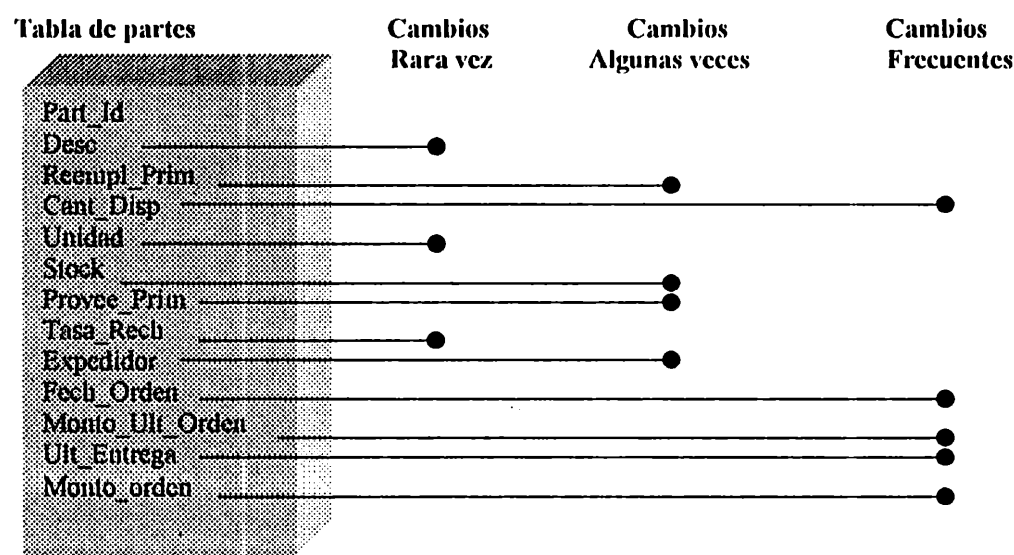




El modelo de datos corporativo es el punto de partida para construir un modelo separado para el ambiente operacional, para dicho ambiente se le agregarán factores de performance al modelo de datos corporativo, en general son muy pocos los cambios que se realizan al modelo corporativo cuando se usa en el ambiente operacional. Sin embargo hay una cantidad importante de cambios que se realizan cuando se aplica al data warehouse. La primera tarea que se realiza es quitar los datos usados con propósitos netamente operativos. Luego se expanden las estructuras de claves del modelo de datos corporativo con un elemento de tiempo. Se agregan datos derivados al modelo de datos corporativo donde los datos derivados( calculados ) se calculan una vez, no repetitivamente. Finalmente, las relaciones de datos se convierten en *artifacts*, técnicas de diseño que sirven para representar la integridad referencial en el Data warehouse.

Una actividad de diseño final cuando pasamos del modelo de datos corporativo al modelo de datos del data warehouse es realizar un análisis de estabilidad que consiste en el agrupamiento de atributos de datos basado en su tendencia al cambio.

En la siguiente figura vemos que como se agrupan atributos de acuerdo a su frecuencia de cambio, esto es, aquellos que cambian rara vez, algunas veces y aquellos que cambian frecuentemente. El resultado neto del análisis de estabilidad que es el último paso de la modelización de datos antes de entrar en el diseño de la base de datos física es crear grupos de datos con características similares.



De este ejemplo se crean tres tablas que se forman de una tabla de propósito general basadas en el análisis de estabilidad realizado.

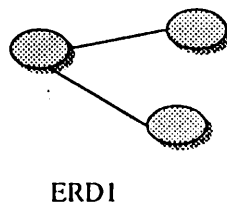
### 3.3 El modelo de datos

Hay tres niveles en la modelización de datos: la modelización de alto nivel (llamada ERD o nivel de entidades y relaciones), la modelización de nivel medio (llamada DIS, conjunto de ítems de datos) y la modelización de bajo nivel (llamada modelo físico).

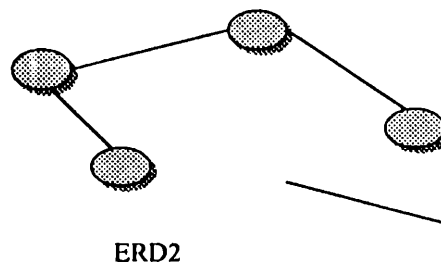
La modelización de alto nivel caracteriza entidades y relaciones, el nombre de la entidad se encierra en un ovalo. Las relaciones entre entidades se dibujan con flechas, la dirección y el número de cabezas de las flechas indican la cardinalidad de la relación y además se indican solamente las relaciones directas. Las entidades que se muestran en el ERD son del más alto nivel de abstracción.

El diagrama de entidades y relaciones (ERD) corporativo es una composición de muchos ERDs individuales que reflejan las diferentes vistas de las personas o comunidades dentro de la corporación, cada comunidad tiene su modelo de datos de alto nivel separado y que en conjunto forman el ERD corporativo.

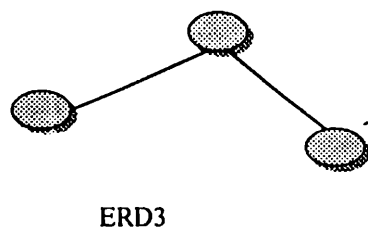
Vista de usuario 1



Vista de usuario 2

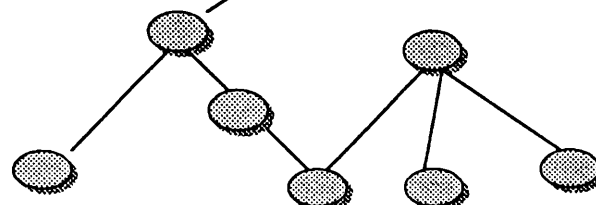


Vista de usuario 3



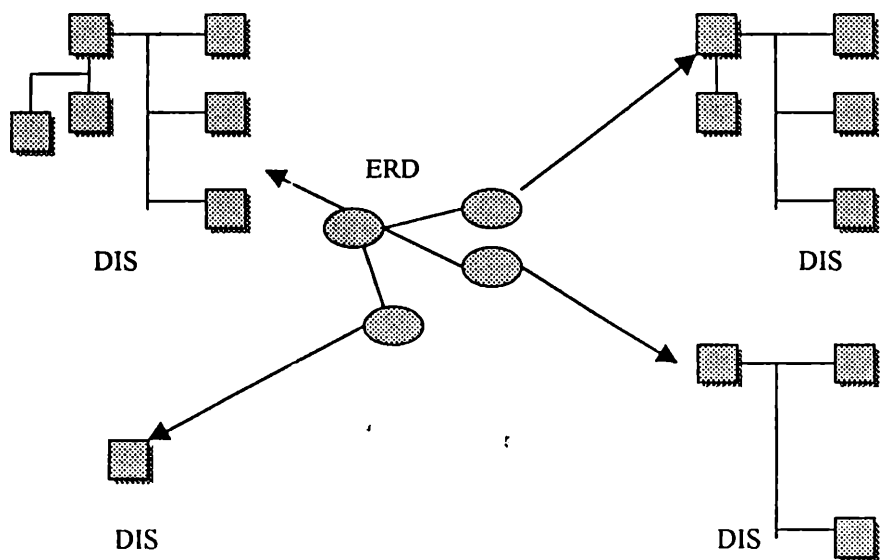
Vista de usuario n

ERDn

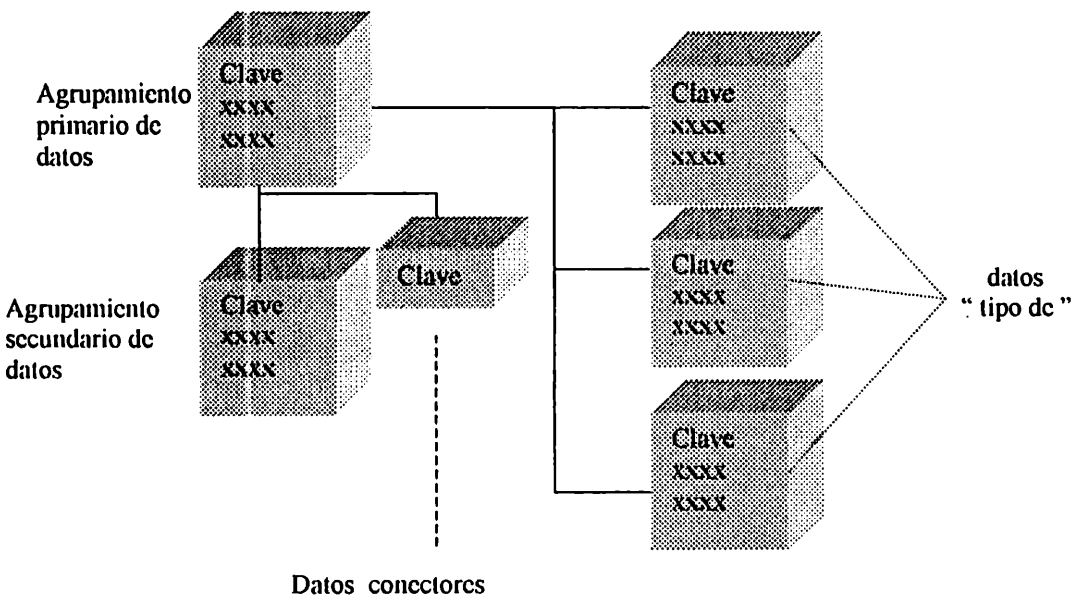


### 3.4 El modelo de datos de nivel medio

Después de haber creado el modelo de datos de alto nivel se crea el modelo de nivel medio o DIS. Para cada entidad identificada en el modelo de datos de alto nivel se crea un modelo de nivel medio. En la siguiente figura el modelo de datos de alto nivel tiene cuatro entidades, para cada una de ellas se desarrolla su modelo de nivel medio.



Los constructores utilizados en la modelización de datos de nivel medio se muestran en la siguiente figura.



Hay cuatro constructores básicos para el modelo del nivel medio:

- Un agrupamiento primario de datos
- Un agrupamiento secundario de datos
- Un conector que identifica las relaciones de datos entre las áreas objetivos importantes
- Datos “ tipo de “

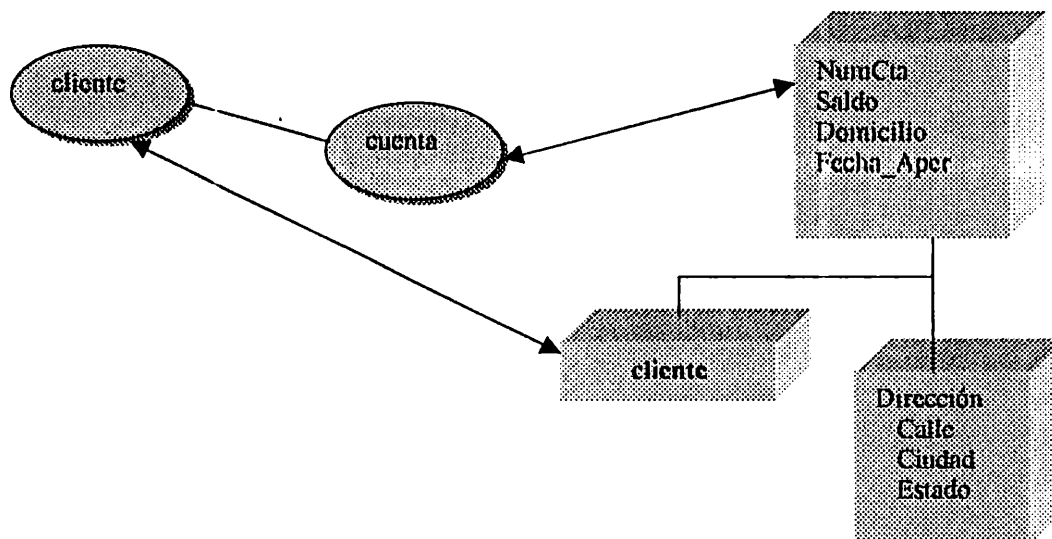
El agrupamiento de datos primario existe una vez y solamente una vez para cada área objetivo, contiene atributos que existen solo una vez para cada área, contiene atributos y claves.

El agrupamiento de datos secundario contiene atributos que pueden existir múltiples veces para cada área objetivo. Este tipo de agrupamiento se indica mediante una línea que sale hacia abajo del agrupamiento primario. Pueden existir tantos agrupamientos secundarios como grupos de datos puedan ocurrir múltiples veces.

El tercer constructor es el conector. El conector relaciona datos de un agrupamiento a datos de otro.

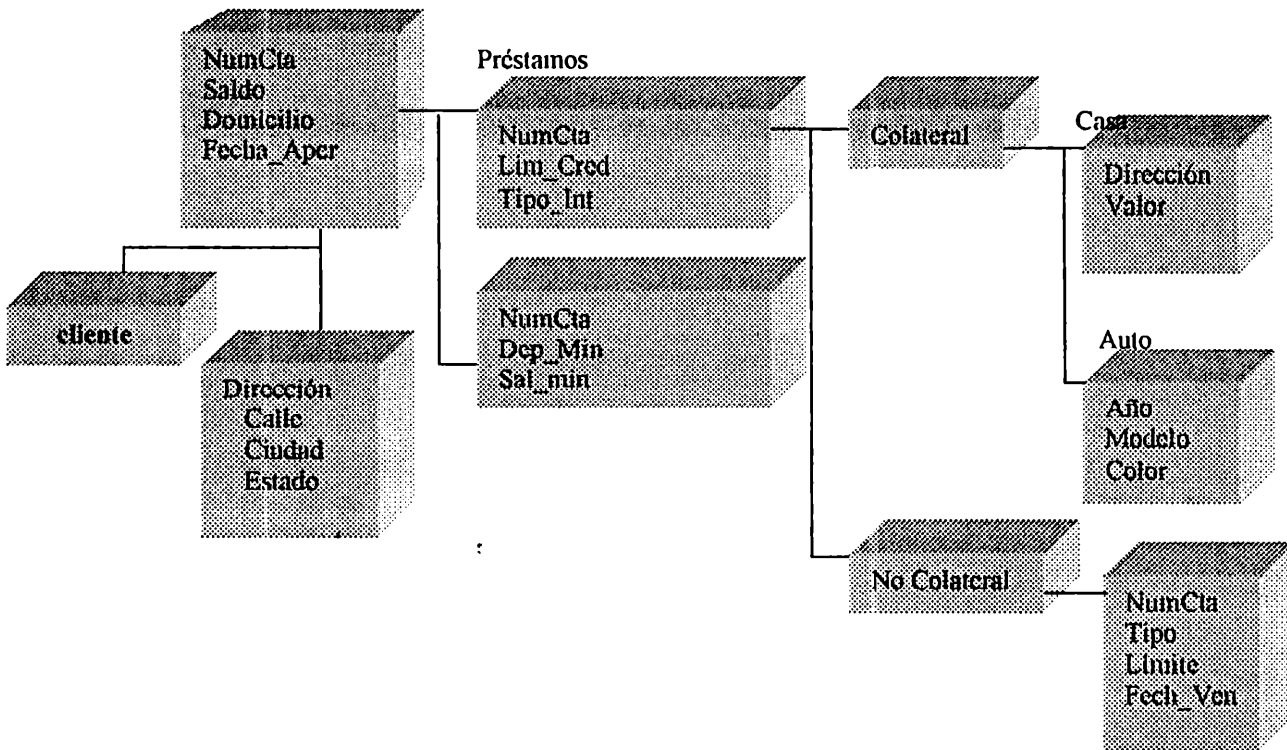
El cuarto constructor en el modelo es el dato de “*tipo de*” . Los datos “*tipo de*” se indican mediante una línea dirigida hacia la derecha de un agrupamiento de datos. El agrupamiento de datos de la derecha es el super tipo y el de la izquierda es el subtipo.

Estos cuatro constructores se usan para identificar los atributos de datos en un modelo de datos y la relación entre ellos. Cuando se identifica una relación en el modelo de alto nivel dicha relación en el DIS, nivel medio, se manifiesta por medio de un par de relaciones de conector. En la figura siguiente vemos uno de esos pares.

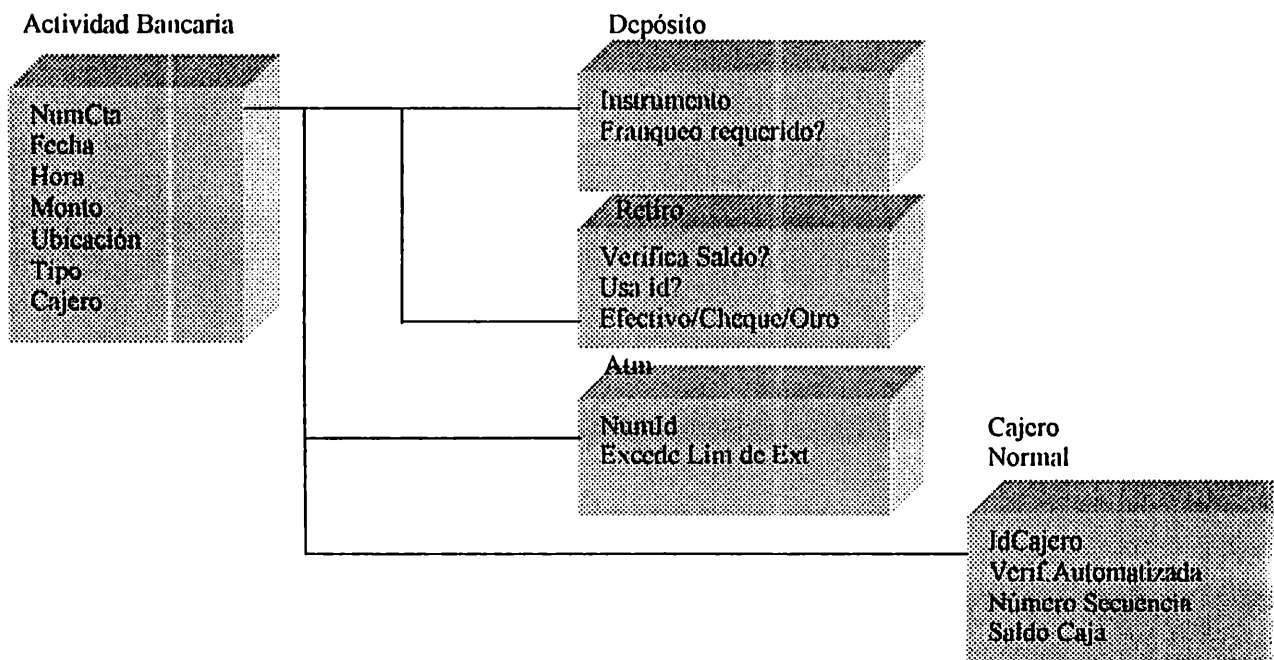


En el ERD se identifica una relación entre cliente y cuenta. En el nivel medio (DIS) de cuenta existe un conector debajo de cuenta, esto significa que una cuenta puede tener varios clientes asignados a ella. En el DIS para cliente habrá un conector a cuenta indicando que cada cliente puede tener una o varias cuentas.

En el próximo gráfico mostramos un ejemplo completo de un DIS de una cuenta para una institución financiera.



En el siguiente gráfico mostramos un ejemplo en el que hay dos criterios “tipo de “. Un criterio “tipo de” es por tipo de actividad, ya sea un depósito o un retiro, el otro criterio indica cajero automático o normal.



Los tipos de actividad representados por este DIS son:

- Depósito automático
- Retiro automático
- Depósito por cajero
- Retiro por cajero

En general cada agrupamiento de datos en el modelo de datos termina siendo una tabla cuando se llega al proceso de diseño de la base de datos.

De la misma manera que los ERDs individuales formaban el ERD corporativo en la modelización de alto nivel, aquí los distintos DIS forman el DIS corporativo.

### **3.5 El modelo de datos físico**

El modelo de datos físico se crea partiendo del modelo de datos de nivel medio extendiéndolo para incluir claves y características físicas al modelo. Antes de asegurar que las tablas que resultan de este proceso están preparadas para ser encastradas en el diseño de la base de datos físico hay que tener en cuenta los aspectos de la granularidad y el particionamiento de los datos que influyen en la performance, además a la estructura de clave se le agrega el elemento del tiempo relevante para cada unidad de datos, luego de esto hay que considerar otros aspectos de diseño físico que describiremos más adelante.

### **3.6 Normalización y desnormalización**

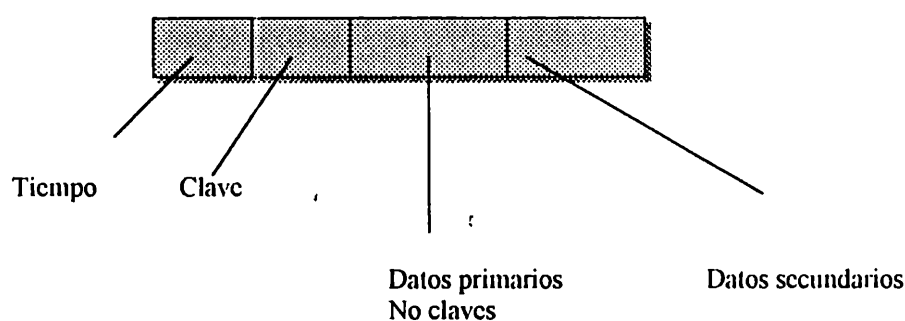
El resultado del modelo de datos es un conjunto de tablas que contienen claves y atributos, cada una de ellas contiene una cantidad de datos pequeña. El trabajo que un programa tiene que hacer para interconectar las tablas dinámicamente es leer una tabla, luego otra tabla y así sucesivamente, el programa tiene que saltar entre tablas.

Cada vez que el programa salta entre las tablas se consumen recursos de entrada/salida ya sea acceso a los datos como acceso al índice para encontrar los datos. Cuando muchos programas están consumiendo recursos, la performance cae, esto es lo que ocurre cuando se crean muchas pequeñas tablas con poca información normalizada.

Una técnica de diseño importante es la introducción de redundancia en las tablas. Un ejemplo clásico de redundancia sería agregar como dato redundante la descripción de un producto en los lugares donde se la use, de esta forma se gana en performance ya que cada vez que se la necesita no hay que estar saltando a la tabla normalizada donde está la descripción. Se podría argumentar que esto produce problemas también de performance en la actualización de este tipo de atributos, pero en el ambiente del Data warehouse la actualización no cumple ningún rol.

### 3.7 Snapshots en el data warehouse

El data warehouse se centra en una estructura de datos llamada *snapshot* o fotografía de la información. A continuación vemos los componentes básicos del snapshot:



Los snapshots se crean como resultado de la ocurrencia de un evento. Hay varias clases de eventos que pueden disparar un snapshot, como por ejemplo, la emisión de un cheque, la recepción de un cargamento, etc. todas estas actividades discretas ocurren en forma aleatoria. Otra clase de actividad que dispara un snapshot es el paso regular del tiempo, se dispara un snapshot en un momento particular en el tiempo, como por ejemplo al final del día, al final de la semana, al final del mes. El pasaje regular del tiempo es predecible y no aleatorio.

Los componentes básicos de un snapshot son:

- Una clave
- Una unidad de tiempo
- Datos primarios que se relacionan solamente a la clave
- Datos secundarios que se capturan como parte del snapshot que no tiene relación directa con los datos primarios o clave.

La clave puede ser única o no, puede ser un elemento de dato simple, pero generalmente, está compuesta de muchos elementos de datos que sirven para identificar los datos primarios. La clave identifica el registro y los datos primarios.

La unidad de tiempo se crea en términos de elementos de tiempo, tales como año, mes, día, hora y trimestre. Generalmente la unidad de tiempo se refiere al momento en que ocurrió el evento descrito por el snapshot, ocasionalmente se refiere al momento en que se produce la captura de los datos.

Los datos primarios son los datos no claves que se relacionan directamente a la clave del registro. Por ejemplo supongamos que la clave identifica la venta de un producto, el elemento del tiempo describe el momento en que finalizó la venta, los datos primarios describen que producto se vendió, el precio, condiciones de la venta, el lugar de venta.

Los datos secundarios, si existiesen, identifican otra información capturada en el momento que se grabó el snapshot. La relación existente entre los datos primarios y secundarios del snapshot se la denomina artifact (artefacto de una relación).

### **3.8 Metadatos (Metadata)**

Un aspecto importante del ambiente del data warehouse es el de los metadatos. Los Metadatos son datos acerca de los datos, han sido parte del procesamiento de información desde que existen los programas y los datos, pero en el mundo de los data warehouses adquieren un nuevo nivel de importancia. El uso de los data warehouses es más efectivo con los metadatos. Los metadatos permiten al usuario final o analista del DSS, el sistema de soporte para la toma de decisiones, navegar a través de todas las posibilidades. Generalmente los metadatos se componen de:

- La estructura de datos que conoce el programador
- La estructura de datos que conoce el analista del DSS
- Los datos origen que alimentan el data warehouse
- La transformación de datos cuando pasa al data warehouse
- El modelo de datos
- La relación entre el modelo de datos y el data warehouse
- La historia de las extracciones



### 3.9 Manejo de tablas de referencia en el data warehouse

En la mayoría de los casos, las tablas de referencia no se incluyen como parte del data warehouse. Por ejemplo, supongamos que una compañía tiene algunas tablas de referencia en 1995 y comienza a crear su data warehouse en 1995. El tiempo pasa y se carga mucha información en el data warehouse. Mientras tanto la tabla de referencia se usa operacionalmente y cambia. En 1999 surge una necesidad de ir del data warehouse a la tabla de referencia, pero dicha tabla no se mantuvo históricamente, esto produce resultados incorrectos. Por ésta razón es que el data warehouse necesita incluir datos de referencia como también datos normales. A continuación describimos dos técnicas de diseño para el manejo de datos de referencia en el data warehouse.

La primera opción de diseño sugiere tomar un snapshot de una tabla de referencia completa cada seis meses. Esta opción es muy buena porque es simple pero es lógicamente incompleta. Supongamos que en el medio del lapso de los seis meses se produjo algún cambio, se agregó una entrada y luego de un mes se borró dicha entrada, tal cambio no se verá reflejado al tomar el snapshot siguiente.

La segunda opción es tomar un primer snapshot de la tabla de referencia y luego durante todo el año registrar todas las actividades de la tabla de referencia de manera que se pueda reconstituir en cualquier momento. Esta segunda opción es lógicamente más completa pero por supuesto es más compleja.

La conclusión con respecto a las tablas de referencia es que no se puede obviar el manejo de las tablas de referencia en el data warehouse, habrá que decidir la técnica a usar, pero alguna hay que usar, ya que las tablas de referencia deben ser manejadas como parte normal del data warehouse.

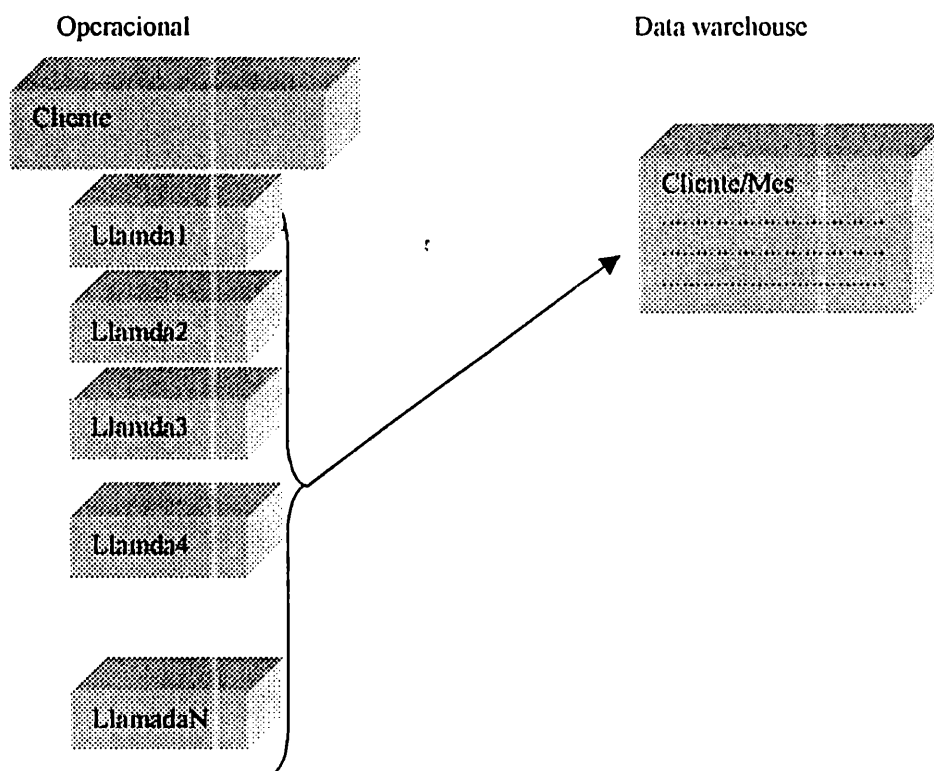
### 3.10 Ciclicidad de datos

La ciclicidad es el tiempo que tarda en reflejarse en el ambiente del data warehouse un cambio de datos producido en el ambiente operacional. La pregunta sería ¿cuál debería ser la longitud de tiempo para reflejar el cambio? Como regla general, deberían pasar al menos 24 horas para reflejarlo en el data warehouse. Hay varias razones para dejar una brecha de tiempo, la primer razón es que cuanto más corto es el tiempo más costosa y compleja debe ser la tecnología, reflejar el cambio en 24 horas es posible fácilmente con tecnología convencional. Otro beneficio de tener 24 horas de tiempo es que da la posibilidad de que los datos se consoliden antes de transferirlos al data warehouse.

### 3.11 Registros perfiles

En algunas ocasiones hay demasiado volumen de información en el ambiente operacional, en otros casos la información cambia muy frecuentemente y en otros casos no hay necesidad comercial de tener un detalle histórico tan meticuloso de los datos. En estos casos se pueden crear registros perfiles o de agregación. Un registro perfil agrupa muchas ocurrencias detalladas distintas de datos operacionales en un solo registro.

Los registros perfiles son disparados por la ocurrencia de algún evento, ya sea una actividad comercial o el paso regular del tiempo. En el siguiente gráfico vemos la actividad de llamadas telefónicas en el ambiente operacional que a fin de mes se agrupan en un solo registro que va al data warehouse.



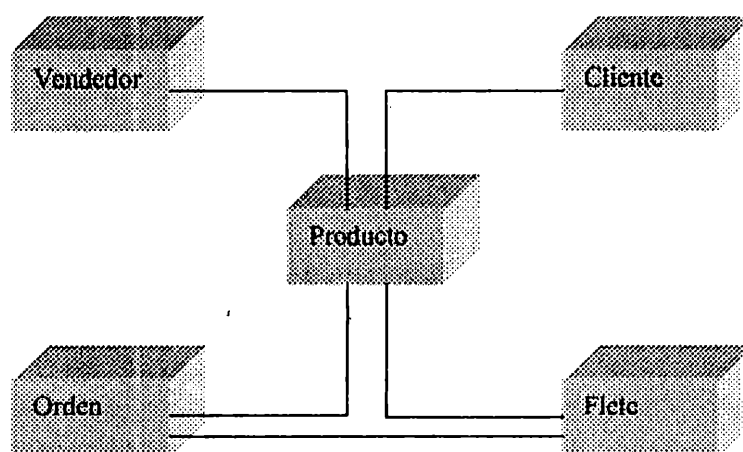
La agregación de datos operacionales en un único registro del data warehouse puede tener muchas formas, por ejemplo:

- Se pueden sumarizar los valores tomados de los datos operacionales
- Se pueden procesar las unidades de datos para encontrar el mayor, el menor, el promedio y así sucesivamente
- Se pueden medir ciertos tipos de datos que caen en ciertos rangos

La creación de registros perfiles se puede extender a la creación de múltiples registros perfiles, perfil de clientes, perfil de tráfico, perfil de análisis para alimentar al data warehouse en distintos áreas de interés, luego estos registros serán usados por usuarios con distintas necesidades de información.

### 3.12 Modelo Dimensional: Esquemas estrella

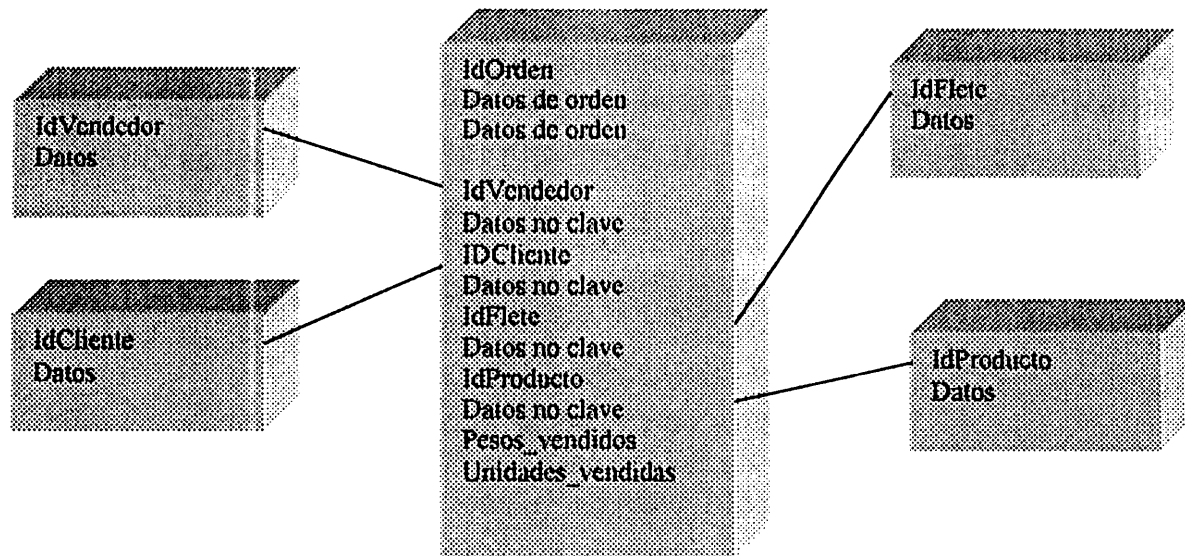
Hay algunas dificultades para poder considerar al modelo de datos como fundamento de diseño del data warehouse. Consideremos el siguiente modelo de datos:



El modelo de datos muestra una figura con cuatro entidades relacionadas entre sí.

Teniendo en cuenta el modelo como se presenta se podría inferir que todas las entidades son iguales, en otras palabras el modelo de datos hace aparentar que todas las entidades son pares entre sí. El modelo de datos produce un efecto chato, plano de las entidades. En realidad, las distintas entidades en el data warehouse se poblarán en formas dispares, por ejemplo las tablas Vendedor, Producto, Flete se poblarán con pocos datos, en forma esparcida, mientras que la tabla Orden será densamente poblada. Si llevamos ésta situación a una representación en tres dimensiones vemos que las entidades no son iguales. Debido al gran volumen que poblará la entidad Orden, dicha entidad necesita un tratamiento de diseño distinto.

El modelo de datos necesario para manejar gran volumen de información es el *modelo dimensional* o *Star Join* (Esquema o Unión Estrella). Para esquematizarlo, volvamos al ejemplo anterior y veamos como se vería, teniendo en cuenta que la entidad Orden estará densamente poblada.



Cada una de las entidades que rodean a la entidad Orden tendrá solamente una cantidad de datos pequeña. El centro de la estrella, la entidad Orden se llama “fact table” o tabla de hechos, las entidades que la rodean se las denomina “tablas dimensión”.

La tabla de hechos o fact table es en donde se almacenan los datos numéricos del negocio, cada uno de éstos es tomado en la intersección de todas las dimensiones. Ejemplo de estos hechos son las Unidades\_vendidas y pesos\_vendidos en la Orden que se refieren a la cantidad de un producto en particular vendido por un vendedor a un cliente y entregado por un flete. Estos hechos son **numéricos**, **continuamente valuados** (puede tomar cualquier valor en un amplio rango de valores) y **aditivos** (pueden ser sumados), aunque puede haber hechos semiaditivos y no aditivos.

Las relaciones de claves foráneas pueden estar acompañadas de información foránea no clave, si esta última se usa frecuentemente en el procesamiento junto a los datos de la *fact table*. Como ejemplo, la descripción de un producto puede estar almacenada en la fact table junto con el número de producto si dicha descripción se usa frecuentemente como parte del procesamiento de la Orden. Pueden existir muchas relaciones de claves foráneas a las tablas de dimensión. Se crea una relación de clave foránea cuando existe la necesidad de examinar la información de la clave foránea junto con datos de la tabla de hechos o fact table. La fact table tiene una clave primaria que es una combinación de las claves foráneas y es llamada clave compuesta o clave concatenada.

En las tablas dimensión es en donde se almacenan las descripciones textuales de las dimensiones del negocio, cada una de estas descripciones ayuda a describir un miembro de la dimensión respectiva, por ejemplo, cada registro en la dimensión producto representa un producto específico. Los atributos de las tablas dimensión

son **textuales y discretos**, su rol clave es servir como fuente de *constraint* (restricción) en una consulta y como cabeceras de columnas en el conjunto respuesta. Un conjunto respuesta típico a una consulta podría ser:

Marca	Pesos Vendidos	Unidades Vendidas
Axon	780	263
Framis	1044	509
Widget	213	444

Esta consulta encontró las marcas de productos vendidos en el primer semestre de 1995 y las presentó al usuario junto a las unidades y pesos vendidos. Para construir este reporte el usuario seleccionó el atributo Marca de la dimensión Producto, Pesos\_vendidos y Unidades\_vendidas de la fact table y los colocó como cabeceras de columna, finalmente especificó la *constraint* “1º Cuatrimestre de 1995” en la dimensión Tiempo. En este punto, la consulta del usuario está bien especificada, y la herramienta de consulta puede generar el SQL requerido y obtener el conjunto respuesta de arriba.

En un data warehouse toda tabla que exprese relaciones muchos-a-muchos debe ser una fact table, las otras tablas son tablas dimensión. Los hechos en la fact table varían en cada carga, los atributos de las tablas dimensión describen el ítem en la dimensión, no varían por la influencia de otra dimensión y son constantes en el tiempo.

Los joins en un esquema dimensional en un data warehouse juegan un rol más importante que en un esquema de entidades relaciones. Cada join expresa una relación fundamental entre ítems del negocio, el join producto, por ejemplo, representa la relación entre la lista maestra de productos y las ventas de la compañía.

En todos los joins en una base de datos dimensional, las claves que implementan esos joins son objeto de administración y atención por el MIS. Esto contrasta con muchos de los joins en una base de datos de entidades relaciones diseñada para OLTP, donde los joins son usualmente claves numéricas generadas artificialmente que tienen poco significado administrativo en la compañía. En una base de datos dimensional, probablemente haya gente para administrar los dos extremos del join, una persona mantiene el archivo maestro de productos y controla la generación de nuevas claves, y la otra persona se asegura de que cada registro de venta contenga claves de producto válidos, por esta razón llamamos *joins MIS* a los joins en un data warehouse.

Las constraints de aplicación se aplican a tablas dimensión individuales, el usuario crea una restricción de aplicación haciendo *browsing* de la dimensión, mirando en los posibles valores y mirando en las relaciones entre los atributos. Raramente tiene

sentido y raramente es posible aplicar una restricción de aplicación simultáneamente a través de dos dimensiones, por eso se linkean las dos dimensiones, que en un diseño dimensional se hace sólo a través de la fact table. Sí es posible aplicar una restricción de aplicación a un hecho en la fact table, que puede ser visto como un filtro sobre los registros.

En una consulta a una base de datos dimensional, primero son evaluadas las restricciones de aplicación dimensión por dimensión; cada dimensión produce un conjunto de claves candidatas, que luego son ensambladas en claves compuestas para ser buscadas en la fact table. Todos los registros encontrados son agrupados y procesados de acuerdo a las especificaciones de la consulta.

El uso de una base de datos dimensional en un data warehouse será el siguiente: El usuario comienza poniendo restricciones sobre las dimensiones haciendo browsing sobre cada una de ellas. El browsing es para propósitos de información sólo para permitir al usuario armar las restricciones correctas sobre cada dimensión. El usuario también “arrastra” cabeceras de fila desde las tablas dimensión y hechos desde la fact table al área del conjunto respuesta (el reporte), entonces inicia el join multi-tabla. El DBMS agrupa y sumaliza cientos, miles o millones de registros de bajo nivel desde la fact table en pequeños conjuntos respuesta y retorna el conjunto respuesta al usuario. Con esto vemos que los atributos en las tablas dimensión juegan un rol vital en el data warehouse, ya que son la fuente de las restricciones y cabeceras de filas de los conjuntos respuesta de los usuarios, esto significa que el data warehouse es tan bueno como lo sean sus atributos de dimensión.

El beneficio de la creación del esquema estrella es simplificar los datos para el procesamiento del sistema de soporte para la toma de decisiones (DSS). Mediante la preunión y la creación de redundancia selectiva el diseñador simplifica los datos para el acceso y análisis que es exactamente lo que necesita para el Data warehouse. Cabe aclarar que si este esquema se usase fuera del ambiente de data warehouse habría muchos inconvenientes ya que afuera ocurren muchas actualizaciones y las relaciones entre tablas se manejan al segundo, sería bastante difícil y pesado de mantener esta estructura. Por la naturaleza del ambiente del data warehouse que es un ambiente de carga y acceso, que contiene información histórica y donde se manejan grandes volúmenes de información, es que la estructura de star join o esquema estrella resulta ideal.

El esquema estrella y el modelo de datos son útiles como técnicas de diseño en el data warehouse, el modelo de datos se aplica como fundamento de diseño a las entidades no voluminosas mientras que el esquema estrella se aplica a las entidades voluminosas, es decir, que el modelo de datos se aplica a las tablas de dimensión y el star join se aplica a la tabla central de la estrella o fact table.

Modelo dimensional es un nuevo nombre para una vieja técnica de bases de datos simples y extensibles. Cuando una base de datos puede ser visualizada como un *cubo* de 3., 4, o más dimensiones, la gente puede imaginar *rebanar* ese cubo por cada una de sus dimensiones.

Toda dimensión cuyos registros definan un punto en el espacio es capaz de soportar múltiples jerarquías geográficas independientes. Una jerarquía geográfica puede ser definida por entidades geográficas referidas a un punto en el espacio, por ejemplo, país, provincia, ciudad, localidad o por un código postal de 5 dígitos.

### 3.12.1 El significado del drill down

Hay una tendencia en los vendedores de herramientas de bases de datos relacionales en insistir que *drilling down* significa descender en la jerarquía de producto. *Drilling down* en un data warehouse significa “muéstrame más detalle”, usualmente para explicar algo remarcable en los datos, en este contexto vemos que *drilling down* es agregar al reporte cabeceras de filas desde las tablas dimensión y *drilling up* es sacar cabeceras de filas. Por ejemplo, si tenemos el siguiente reporte:

Marca	Paquete	Ventas
Brawny	pack-2	\$ 50
Brawny	pack-3	\$110
Brawny	pack-6	\$ 75

Agregando Color como una cabecera de columna adicional lo expandiremos a:

Marca	Paquete	Color	Ventas
Brawny	pack-2	Blanco	\$ 20
Brawny	pack-2	Marrón	\$ 60
Brawny	pack-3	Blanco	\$ 35
Brawny	pack-3	Verde	\$ 45
Brawny	pack-3	Marrón	\$ 75
Brawny	pack-6	Marrón	\$110
Brawny	pack-6	Rosa	\$115

Con esto vemos que los productos de color marrón fueron los más vendidos y que es cuestionable seguir vendiendo los productos de color blanco. Agregar Color como un paso de drill down fue la clave en este análisis de negocio.

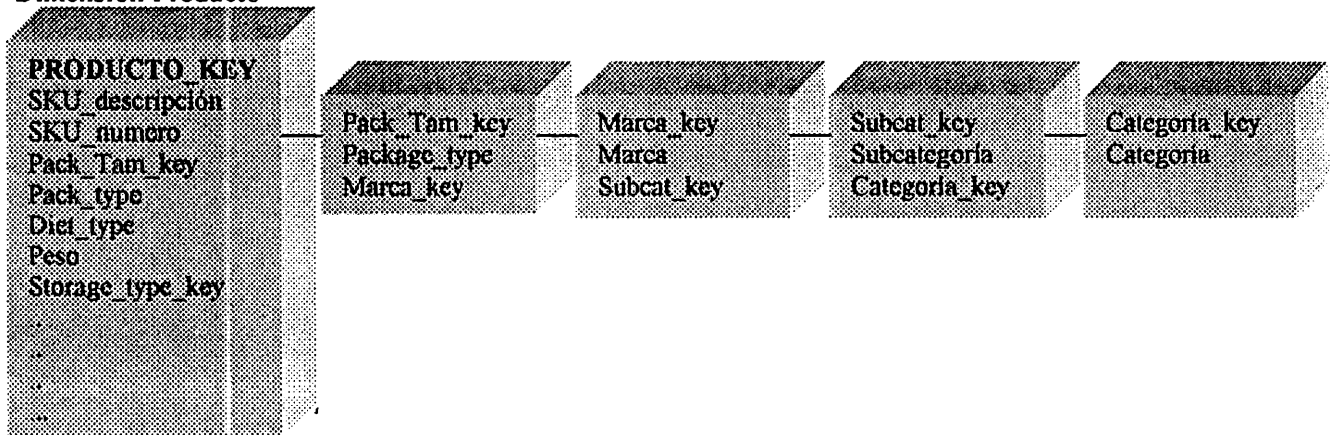
Una dimensión típica contiene una o más jerarquías naturales junto con otros atributos que no tienen una relación jerárquica con algún atributo de la dimensión.

Cualquier atributo perteneciente o no a una jerarquía puede ser libremente utilizado en drilling down o drilling up.

### 3.12.2 Snowflake

Dimensiones con jerarquías son a menudo descompuestas por diseñadores en una estructura denominada *snowflake* (copo de nieve), por ejemplo:

Dimensión Producto



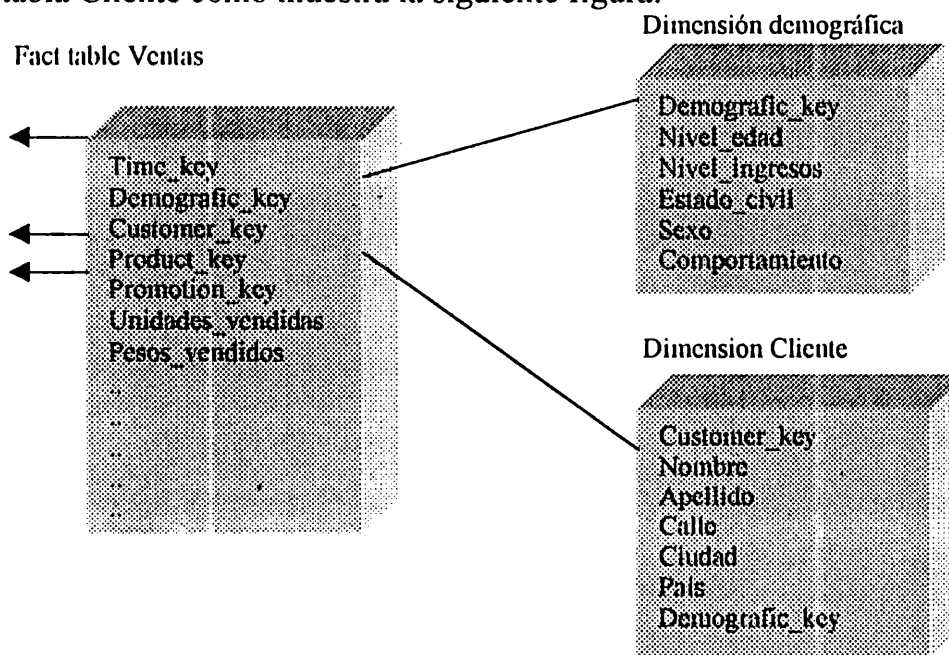
Cada una de las relaciones muchos\_a\_uno son tratadas como una tabla separada. La razón para usar esta estructura es ahorrar almacenamiento. En una gran tabla de productos de 100 MB, es posible ahorrar varios MB no copiando los campos de texto más repetidos cientos de veces; aunque en realidad el ahorro es irrelevante y además degrada la performance del *browsing*. Recordemos que el *browsing* es el acto de navegar en una dimensión para tener un entendimiento intuitivo de cómo los atributos se relacionan entre sí o para construir una constraint sobre la dimensión. Si una tabla ha sido dividida en un *snowflake*, y se intenta hacer *browsing* entre atributos separados, posiblemente en varias estructuras de árbol, es inevitable que la performance esté comprometida, o aún destruida. Lo mejor para el *browsing* de propósito general en una gran tabla dimensión es la implementación de una tabla plana, para preservar la mejor performance y presentar una vista más simple al usuario final. La única excepción defendible a la prohibición del snowflake es cuando, por ejemplo, la dimensión Cliente tiene un gran bloque de datos demográficos repetidos.

Los campos demográficos son los campos que hemos usado más ampliamente como edad, sexo, cantidad de hijos, nivel de ingresos, nivel de educación y comportamiento relacionado a la compra o utilización de un crédito. Además estos campos los hemos agrupado para seleccionar un subconjunto interesante de cliente base o tipo. La técnica más efectiva para usar estos atributos demográficos en una



gran dimensión Cliente es separar en uno o más conjuntos de estos atributos en minidimensiones demográficas. Aunque lógicamente estos atributos permanecerán en la dimensión Cliente primaria podemos ganar en performance creando las minidimensiones. Si cinco o seis de las variables demográficas son aisladas en una tabla separada sólo necesitamos almacenar las distintas combinaciones de atributos demográficos.

Una minidimensión demográfica puede ser vinculada directamente a la fact table o a la tabla Cliente como muestra la siguiente figura:



Las clave demográficas pueden existir como una clave foránea en la fact table y en la dimensión Cliente. Pueden también ser usadas directamente con la dimensión Cliente cuando los atributos demográficos necesitan ser examinados interactivamente con los atributos de la dimensión Cliente. Los atributos demográficos también pueden ser usados con la dimensión Cliente para producir listas de mailing.

### 3.12.3 Dimensiones lentamente cambiantes

Hasta este punto en el diseño de bases de datos multidimensionales hemos asumido que las dimensiones como Cliente y Producto son independientes del tiempo, pero en el mundo real esto no es cierto. Con el tiempo las descripciones y formulaciones de productos reales evolucionan lentamente. En particular, los clientes son constantemente cambiantes, las personas cambian sus nombres, su estado civil, tienen más hijos y modifican sus direcciones. Hemos visto hasta ahora que una de

las responsabilidades del data warehouse es representar el estado de la historia, por lo tanto, debemos explotar el hecho de que muchas dimensiones son “casi” constantes en el tiempo y que podemos preservar la estructura dimensional independiente con sólo agregados relativamente menores para capturar la naturaleza cambiante en el tiempo. Llamamos dimensiones cambiantes lentamente a este tipo de dimensiones. Cuando tenemos dimensiones de este tipo debemos decidir entre tres tipos de opciones para seguir la pista de los cambios:

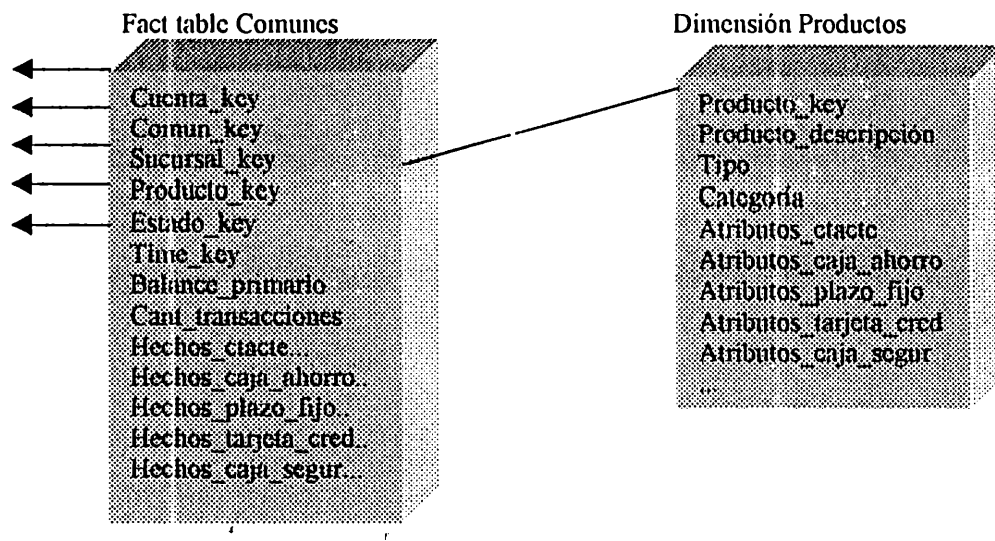
- Sobrecribir los valores viejos en el registro de la dimensión perdiendo la posibilidad de seguir la historia. La ventaja de esto es que es lo más fácil de implementar ya que no es necesario algún cambio en otro lugar en el registro y no afecta a claves en la base de datos.
- Crear un registro adicional en el momento del cambio con los nuevos valores de atributos y segmentar la historia entre la vieja y la nueva descripción. Esta opción requiere que la clave dimensión sea generalizada, puede ser suficiente tomar la clave de producción y agregar dos o tres dígitos de versión al final de la clave para simplificar el proceso de generación de claves. La creación de claves generalizadas usualmente es responsabilidad del equipo de desarrollo del data warehouse, y siempre requiere de metadatos para seguir la pista de las claves generalizadas que ya han sido usadas.
- Crear nuevos campos “corrientes” dentro del registro de dimensión original para registrar los nuevos valores de atributo mientras se mantienen los valores originales también, con lo cual se puede describir la historia hacia adelante y hacia atrás desde el cambio en términos de los valores originales del atributo o en términos de valores corrientes.

Llamamos a estas opciones dimensiones lentamente cambiantes Tipo Uno, Tipo Dos y Tipo Tres.

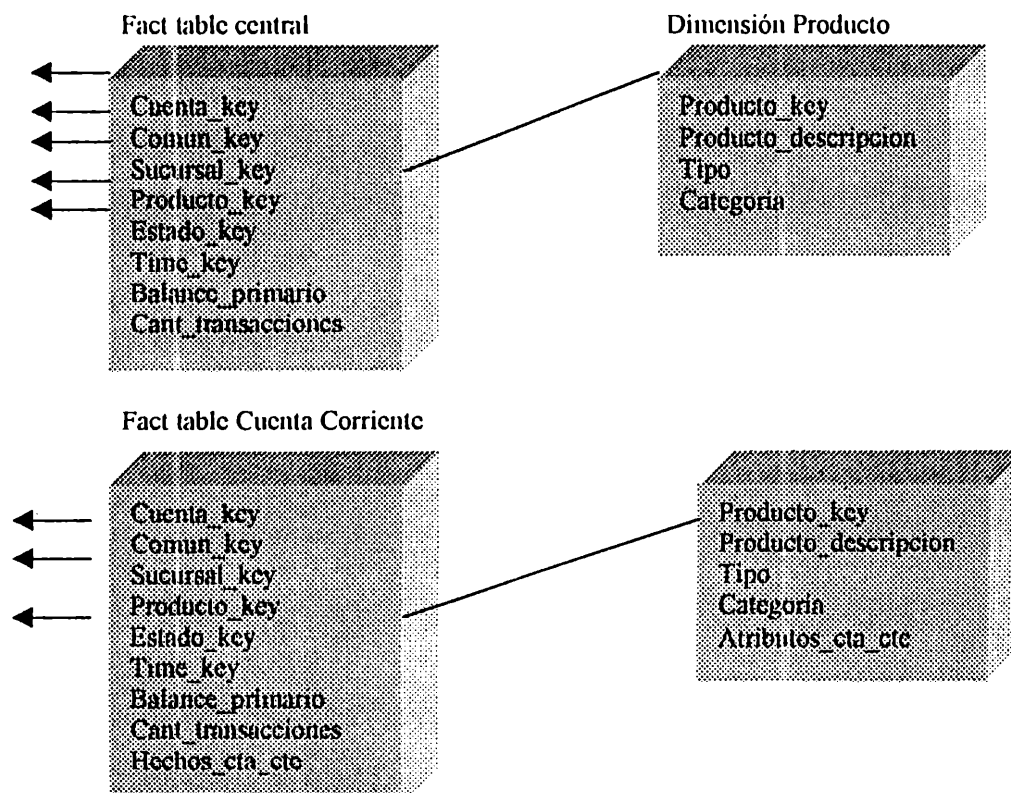
### 3.12.4 Productos Heterogéneos

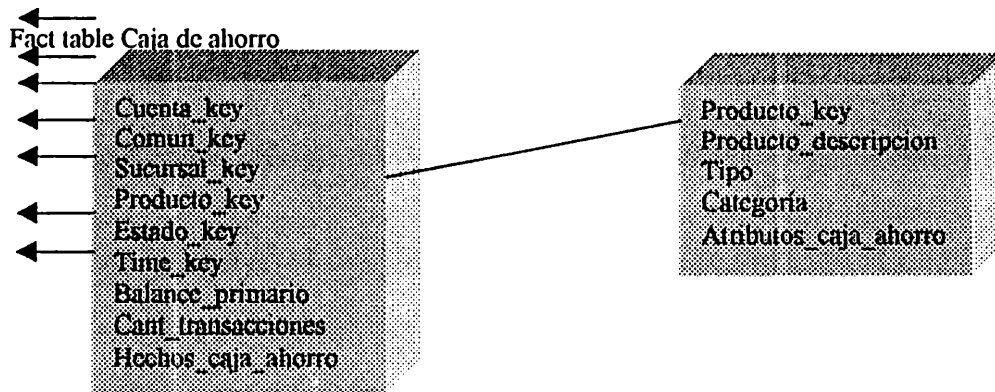
En un data warehouse donde una dimensión debe describir una gran cantidad de ítems heterogéneos, la técnica recomendada es crear una fact table y una dimensión centrales (*core*) para permitir las consultas a través de los tipos dispares, y crear una fact table y una dimensión a medida (*custom*) para consultar cada tipo individual. Por ejemplo, cada tipo de cuenta en una banco tiene un balance primario asociado, pero cada una de ellas tiene atributos especiales que no son compartidos por todas; por ejemplo las cuentas de cheques tienen balances mínimos, límites de giro en descubierto, cargos y vínculos a tarjetas de crédito. Las cajas de ahorro tienen

además intereses pagados, opciones de depósito directo y débitos, los plazos fijos tiene menos cosas en común con las cuentas corrientes y cajas de ahorro, pero tienen fechas de renovación, fecha de vencimiento, tasa de interés; las tarjetas de créditos tienen niveles de créditos, pagos, número de tarjetas entregadas y número de meses de mora; las cajas de seguridad tienen tipo de caja y precio. Si incluimos todo en una sola tabla dimensión y en una fact table nos quedaría:



Utilizando la técnica descripta, creamos la fact table central con los balances primarios y la cantidad de transacciones, y una tabla dimensión producto central con la información de la jerarquía del producto. También creamos la fact table y tablas dimensión a medida como muestra la siguiente figura:





El diseño completo debería tener un subesquema a medida para cada tipo de producto. Aunque esto se ve complejo, sólo el DBA ve todas las tablas a la vez.

Desde la perspectiva del usuario, un análisis es un análisis de producto cruzado, por lo tanto la atención estará enfocada en la fact table y dimensión central, o es el análisis de un producto particular, y la atención se enfocará en los subesquemas de fact table y dimensiones a medida. En general., no tiene sentido combinar datos desde dos o más subesquemas a medida porque, por definición, los hechos y atributos son disjuntos.

Las claves para las tablas de dimensión Producto a medida son las mismas claves usadas en las tablas de dimensión central, las cuales contienen todas las claves de producto posibles. Cada tabla de dimensión producto a medida es un subconjunto de la tabla dimensión producto central, y contiene atributos de producto extra específicos al tipo de producto.

## Capítulo 4

# LA TECNOLOGÍA Y EL DATA WAREHOUSE

Como vimos anteriormente un Data Warehouse lo definimos como un *Almacén de Datos* el cual reúne los datos de los diferentes sistemas de una empresa; integrándolos en este almacén, los cuales son usados posteriormente a través del analista DSS para lograr sus objetivos. Por lo tanto existen requerimientos tecnológicos para el desarrollo de un proyecto de Data Warehousing entre los cuales podemos citar los siguientes :

- *(Almacenar gran cantidad de información)*

Como ya vimos anteriormente el *Almacén de Datos* se puebla de los datos de históricos y corrientes de los diferentes sistemas de la organización los cuales conforman un gran volumen de información que el esquema debe manipular de forma eficiente para poder satisfacer los posteriores análisis de la información.

- *(Manejar múltiples medios de almacenamiento)*

Por la gran cantidad de datos que residen en un Data Warehouse, la tecnología utilizada para implementar el mismo debe permitir diferentes medios de almacenamiento, los cuales se pueden clasificar en términos de velocidad de acceso y costo de almacenamiento. Entre ellos podemos citar :

- la memoria principal que en términos de acceso es la mas rápida y con respecto al costo es la más cara
- almacenamiento en discos ópticos es cual es menos veloz que la memoria y a su vez menos caro también.
- otros medios serían memorias cache y expandidas las cuales son mas rápidas que los discos y más caras también.

- *(Mecanismos de búsqueda a través índices y mecanismos de monitoreos)*

Debido a que no sabemos de antemano que datos, los analistas, recorrerán por lo tanto para acelerar esta búsqueda se debe permitir utilizar índices. Los diferentes técnicas de indexación que debe soportar son :

- *( Índices secundarios)*



- ( Índices dinámicos
- Índices temporarios )

Hay muchas razones para mantener un mecanismo de monitoreo eficiente, las cuales los podremos identificar de la siguiente forma :

- determinar cuando una reorganización es necesaria
- determinar cuando existen demasiado overflow de datos
- determinar si existe espacio disponible suficiente para operar

Todos estos mecanismos apuntan hacia la performance del esquema de Data Warehouse implementado en tanto a tiempos de respuesta y dinamismo de búsqueda.

- *(Soportar interfaces desde y hacia otras tecnologías)*

Otro importante aspecto de la tecnología de implementación del Data Warehouse, es la posibilidad de recibir datos desde y enviar hacia otras tecnologías, utilizando mecanismos eficientes y fáciles para su realización. Normalmente este mecanismo es implementado a través de un utilitario dedicado a dicha tarea.

- *(Programar y diseñar la ubicación de los datos)*

Esta ubicación se refiere a nivel de bloques de física información para si economizar tanto espacio de almacenamiento como tiempos de accesos a los datos debido a su alocaión pre-programada, como sería evaluar a priori el posible de espacio de una tabla (definir un tamaño aproximado para su alocaión ) para que la misma resida en posiciones contiguas en el dispositivo de almacenamiento utilizado. Como así también el análisis del crecimiento de la misma durante la vida en el data warehouse.

- *(Soporte para manejo de Metadatos)*

Es uno de los importantes requerimientos tecnológicos pues es el que determina como esta estructurado el esquema de datos en el Data Warehouse. Normalmente los metadatos constituyen :

- Las tablas de estructura del Data warehouse
- Las tablas de atributos
- El esquema de registros utilizados
- El mapeo de los datos desde el sistema de registros hacia el Data Warehouse
- El modelo de datos utilizado
- Rutinas comunes para el acceso a los datos

- *(Lenguaje de manipulación de los datos)*

Un lenguaje de manipulación debe permitir accesos a los datos utilizando el lenguaje de tipo SQL como también un buen manejo de inserciones, actualizaciones y

borrados de datos como así también el uso de índices en la definición de las sentencias de acceso.

- ***Carga de los datos hacia el Data Warehouse***

Como ya sabemos una de las tareas más importantes en un esquema de Data Warehouse es el refresco con los datos de los diferentes sistemas, el cual puede ser realizado a través del lenguaje de manipulación de datos y/o utilizando un utilitario diseñado para ello, el debe tener en cuenta no solo la carga de los datos si no también la creación de los índices. En el tiempo de carga se debe tener en cuenta la integridad de los datos el cual es llevado a cabo mediante los mecanismo de transformación de los mismo.

- ***Utilizar índices eficientemente***

Por el volumen de información que existe en un Data Warehouse, el utilizar índices eficientes permite al DSS (una mayor rapidez en obtener su respuesta), siempre y cuando los mismos se encuentren definidos correctamente, algunas formas de indexación que se recomiendan utilizar son :

- Utilizar índices BIT MAPS
- Tener multinivel de indexación
- Guardar parte o todo el índice en memoria principal
- Compactar las entradas de los índices cuando sea posible
- Crear índices selectivos.

- ***Compactación de los datos***

Esta compactación permite utilizar la (menor cantidad de espacio) Este puede llevarse a cabo en el momento de la inserción o utilizar procesos de monitoreos para realizar este tipo de tareas.

- ***Otros requerimientos tecnológicos que debemos tener en cuenta son :***

- Manejo de claves compuestas en el sistema de índice)
- Longitud variable de los datos, este requerimiento se hereda debido a que como los datos fluyen de diferentes sistemas con esquemas de datos distintos.
- Recuperación rápida ante posibles errores de almacenamientos.)

Es necesario mantener dispositivos de almacenamiento secundarios para poder realizar esta recuperación, lo cual trae aparejado un costo muy alto, pues se deben tener una doble cantidad de dispositivos de almacenamiento para cubrir este requerimiento

## 4.1 Tipos DBMS y el Data Warehouse

Con la llegada del Data Warehouse y el análisis DSS se ha creado un nuevo motor de bases de datos en el cual se optimizaron las funciones para manejar gran cantidad de volúmenes de información como así también el uso más eficientes de índices y tipos de los mismos.

(Algunas diferencias que podemos citar con respecto a los DBMS de ambientes operaciones son las siguientes :

- Permite manejar una gran cantidad de datos (en términos de gigabytes)
- Una importante diferencia es el manejo de actualización de los datos ya existentes; pues para DBMS tradicionales insumen un tiempo muy significativo debido a las operaciones que hay que realizar para mantener la integridad en la base de datos, esta integridad le cuesta al motor un gran tiempo de proceso a diferencia de (los data warehouse los cuales tienen como regla que los mismos no son actualizables) o bien estas actualizaciones que surgen de los errores de procesamiento se realizan en off line.
- Otra diferencia que podemos citar es con respecto a los manejos espacios libres reservados en los bloques para las futuras expansiones en el momento de actualizaciones e inserciones de nuevos datos. Típicamente este espacio es llamado DBMS freespace el cual puede ser hasta del 50 % del bloque de datos. A diferencia del DBMS utilizado por el data warehouse en cual este porcentaje se mantiene en 0 debido a que no se necesitan expansiones físicas de los bloques una vez realizada la carga inicial y como las actualizaciones son poco frecuentes no es necesario tenerlo en cuenta. Pero sí hay que tener en cuenta cuando se realiza por (ejemplo la definición de una tabla) cual es el espacio requerido por la misma, así poder reservar a priori una cantidad de espacio razonable para la misma Ej. : Reservar espacio en un extend de datos para colocar un año de información.
- Una cuarta diferencia es a (nivel indexación de datos). En una DBMS tradicional se encuentra restringido el uso de un número finito de índices, debido a que cuando se realizan tanto inserciones y actualizaciones estas se deben reflejar en los mismos, lo cual influye en el tiempo de respuesta. Mientras que en un ambiente de data warehouse como estos índices son creados solo cuando se incorporan los datos y normalmente esta inserción se realiza en off-line, el tiempo de respuesta pasa un papel secundario, permitiendo así manejar esquemas más complejos de indexación



## 4.2 Bases de datos Multidimensionales y el Data Warehouse

Otra tecnología discutida en los ambientes de data warehouse son las bases de datos multidimensionales las cuales son llamadas *datamart*. Este tipo de bases provee una estructura que permite organizar más flexiblemente los datos y una dinámica de exploración de datos entre sumalizaciones y detalles de los datos. Este tipo de bases de datos están orientados tanto a usuarios finales como analistas DSS.

Existe una interesante y complementaria relación entre este tipo de base de datos y el data warehouse, pues los datos que residen el data warehouse proveen una robusta y conveniente fuente de recursos para la DBMS multidimensional., los cuales refrescan la DBMS multidimensional en forma periódica o bien cuando es necesario, pues estos datos ya se encuentran pre-formateados ( integrados , decodificados y organizados) pues primero ingresaron en el data warehouse, haciendo este refresco más eficiente sobre si tomaríamos los datos para esta base del ambiente operacional.

Muchas veces pensamos que DBMS multidimensionales pueden ser una tecnología óptima para realizar nuestro esquema de Data warehouse, pues no es así. Para ver el error que comentemos en este pensamiento consideremos las siguientes diferencias :

- El data Warehouse, mantiene una gran cantidad de datos, mientras DBMS mantiene menos cantidad pues ya existe en ellos un nivel medio de sumarización
- Las BDMS multidimensionales mantienen un mecanismo de acceso y análisis poderoso e ilimitado conocido como *slice and dice*
- Un Data Warehouse contiene datos históricos (detallados) con un horizonte de tiempo de los 5 a los 10 años de información, mientras que DBMS multidimensionales tiene un horizonte de tiempo menor.

---

## Sección 3

### Metodología

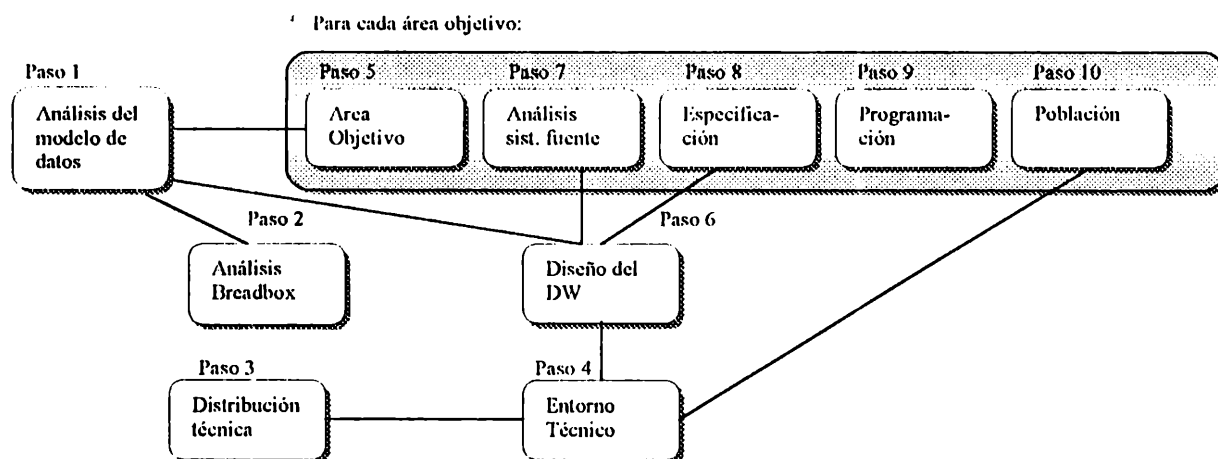
---

Esta sección consta de un solo capítulo, donde desarrollamos una metodología de diseño de Data Warehousing, ésta es del tipo *data driven*. Con la metodología *data-driven* el desarrollo de sistemas se realiza sobre código y datos previamente construidos, para ello éstos deben ser identificados y/o reconocidos (tanto los datos como la relación entre los mismos). La clave de este reconocimiento es obtener el modelo de datos.

## Capítulo 5

### METODOLOGÍA DE DISEÑO

Para la construcción de nuestro data warehouse utilizaremos una metodología de diseño *data-driven*, y para la implementación del modelo de datos utilizaremos el modelo multidimensional. Con la metodología *data-driven* el desarrollo de sistemas se realiza sobre código y datos previamente contruidos, para ello éstos deben ser identificados y/o reconocidos (tanto los datos como la relación entre los mismos). La clave de este reconocimiento es obtener el modelo de datos. La metodología, está conformada por una serie de actividades o pasos que mantienen un cierto orden (existe una actividad precedente y otra siguiente), un tiempo estimado para su realización y una frecuencia de ejecución. Gráficamente:



#### Paso 1: Análisis del modelo de datos.

*Actividad Precedente:* el compromiso de crear un data warehouse.

*Actividad Siguiete:* Análisis del área objetivo, Análisis breadbox, Diseño del data warehouse.

*Tiempo estimado:* varía dependiendo del estado y calidad del modelo.

*Frecuencia de Ejecución:* una vez.

*Objetivo:* Para este paso necesitamos tener:

- Identificadas las áreas objetivo

- Habilidad de acceder a los datos en conjunto o un registro por vez.

- Definidos los límites del modelo
- Separados los datos primitivos de los derivados
- Para cada área objetivo, identificados:
  - Claves
  - Atributos
  - Grupos de atributos
  - Relaciones entre atributos
  - Ocurrencias múltiples de datos
  - Datos “tipo de”

*Parámetros de Éxito:* como resultado de este paso obtenemos el modelo de datos claramente especificado.

## **Paso 2: Análisis Breadbox**

*Actividad Precedente:* Análisis del modelo de datos.

*Actividad Siguiente:* Diseño del data warehouse

*Tiempo estimado:* de un día a dos semanas dependiendo del alcance y del modelo previamente definidos

*Frecuencia de Ejecución:* una vez.

*Objetivo:* encontrar la granularidad de los datos.

*Parámetros de Éxito:* estimación de la cantidad de datos para el horizonte de 1 a 5 años para el data warehouse. Basados en esta estimación, se deciden los diferentes niveles de granularidad.

## **Paso 3: Distribución Técnica**

*Actividad Precedente:* compromiso de construir un data warehouse.

*Actividad Siguiente:* Preparación del entorno técnico.

*Tiempo estimado:* Una semana.

*Frecuencia de Ejecución:* una vez.

*Objetivo:* definir los requerimientos técnicos para manejar el data warehouse.

*Parámetros de Éxito:* la definición técnica debe satisfacer los siguientes criterios:

- Habilidad de manejar grandes cantidades de datos
- Habilidad de permitir que los datos sean accedidos flexiblemente.
- Habilidad de organizar los datos de acuerdo a un modelo de datos.
- Habilidad de enviar y recibir datos a una amplia variedad de tecnologías.
- Habilidad de cargar datos en forma masiva.
- Habilidad de acceder a los datos en conjunto o un registro por vez.

## **Paso 4: Preparación del entorno técnico.**

*Actividad Precedente:* Distribución Técnica

*Actividad Siguiente:.* Diseño del data warehouse, Población.

*Tiempo estimado:* Una semana a un mes.

*Frecuencia de Ejecución:* una vez.

*Objetivo:* identificar técnicamente la arquitectura del data warehouse .

Algunos temas que deben ser tratados aquí son:

- Cantidad de almacenamiento requerido (DASD).
- Conexiones de red requeridas.
- Volumen de procesamiento anticipado
- Cómo minimizar o aliviar conflictos entre programas.
- Volumen y naturaleza del tráfico que será generado desde la tecnología que controla el data warehouse.

*Parámetros de Éxito:* los componentes técnicos que serán instalados para manejar datos incluyen:

- La red
- DASD
- El sistema operativo que maneje el DASD
- La interfaz a y desde el data warehouse
- El software usado para manejar el data warehouse
- El data warehouse.

## **Paso 5: Análisis del área objetivo.**

*Actividad Precedente:* Análisis del modelo de datos

*Actividad Siguiente:.* Análisis del sistema fuente.

*Tiempo estimado:* un día.

*Frecuencia de Ejecución:* una vez por proyecto de población.

*Objetivo:* seleccionar el área objetivo a ser poblada. La primer área objetivo seleccionada debe ser lo suficientemente grande para ser significativa y suficientemente chica para ser implementada.

*Parámetros de Éxito:* es una definición de qué datos serán poblados a continuación. Cuando es realizado correctamente, los objetivos seleccionados satisfacen las necesidades de la etapa corriente de desarrollo del data warehouse.

## **Paso 6: Diseño del data warehouse.**

*Actividad Precedente:* Análisis del modelo de datos, Análisis del sistema fuente, Análisis breadbox.

*Actividad Siguiente:* Especificación del Programa.

*Tiempo estimado:* una a tres semanas.

*Frecuencia de Ejecución:* una vez.

*Objetivo:* diseñar el data warehouse teniendo en cuenta las siguientes características:

- Niveles de granularidad definidos, si existen.
- Orientación de los datos hacia los objetivos principales de la corporación.
- Presencia de datos primitivos y derivados.
- Ausencia de datos no-DSS.
- Varianza en el tiempo de cada registro de dato.
- Desnormalización física de datos donde sea aplicable.
- Creación de *artifacts* de datos que pasan del operacional al data warehouse.

*Parámetros de Éxito:* cuando este paso es hecho apropiadamente, el resultado es un data warehouse que tiene una cantidad de datos manejable que pueden ser cargados, accedidos, indexados y consultados en forma razonablemente eficiente.

## **Paso 7: Análisis del sistema fuente.**

*Actividad Precedente:* Análisis del área objetivo.

*Actividad Siguiente:* Especificación del Programa, Diseño del data warehouse.

*Tiempo estimado:* una semana por área objetivo.

*Frecuencia de ejecución:* una vez por área objetivo.

*Objetivo:* identificar los datos fuentes en el sistema operacional para el data warehouse. Se estudia la integración de datos como:

- Estructura, resolución de las claves de los datos que pasan del operacional al entorno DSS.
- Atributos:
  - Qué hacer cuando hay múltiples fuentes de las cuales elegir
  - Qué hacer cuando no hay fuentes de datos.
  - Qué transformaciones deben ser hechas a los datos que pasan al entorno DSS.
- Cómo será creada la varianza en el tiempo de los datos.
- Cómo será creada la estructura DSS desde la estructura operacional.
- Cómo aparecerán las relaciones operacionales en el entorno DSS.

*Parámetros de Éxito:* el resultado de este paso es el mapeo de datos desde el entorno operacional al entorno DSS.

Para nuestra implementación (modelo multidimensional), el Paso 6 y el Paso 7 se unen en el siguiente grupo de actividades:

- Identificar los procesos más importantes y la estructura de la fact table.
- Granularidad de la fact table
- Elegir grano de las transacciones versus grano por snapshot de tiempo.
- Identificar dimensiones para cada fact table.
- Hechos para la fact table identificando los campos en el sistema operacional.
- Atributos de las dimensiones ( cada campo del operacional identificado).
- Tablas heterogéneas identificadas.
- Atributos cambiantes de las dimensiones.
- Minidimensiones demográficas.
- Dimensiones aggregates iniciales.
- Duración de cada tabla.
- Urgencia de cada tabla.
- Puesta en marcha. ( cuándo puede el usuario usar el data warehouse ).

## **Paso 8: Especificación del programa**

*Actividad Precedente:* Análisis del sistema fuente, diseño del data warehouse.

*Actividad Siguierte:* Programación

*Tiempo estimado:* una semana por proceso de extracción / integración.

*Frecuencia de ejecución:* una vez por cada programa que necesita ser escrito.

*Objetivo:* formalizar la interfaz en términos de especificación de programa, que incluye:

- Cómo sé qué datos operacionales examinar ?
  - Los datos operacionales tienen fecha y hora ?
  - Hay archivos delta ?
  - Hay log de auditoría?
  - Pueden cambiarse los códigos fuente para crear un archivo delta ?
  - Hay archivos antes y después para ser comparados ?
- Cómo almaceno la salida una vez examinada ?
  - Son los datos DSS preformateados ?
  - Son los datos agregados ?
  - Son los datos reemplazados ?
  - Son hechas actualizaciones en el entorno DSS ?

*Parámetros de Éxito:* el resultado de este paso es la especificación del programa que será usado para pasar los datos del entorno operacional al data warehouse.

## **Paso 9: Programación.**

*Actividad Precedente:* Especificación

*Actividad Siguiente:* Población

*Tiempo estimado:* una semana por programa de extracción / integración.

*Frecuencia de ejecución:* una vez por programa.

*Objetivo:* realizar actividades estándares de programación, tales como:

- Desarrollo de pseudocódigo.
- Codificación
- Compilación
- Debugging
- Testeos

*Parámetros de Éxito:* el código generado es eficiente, documentado, fácil de ser cambiado y completo.

## **Paso 10: Población**

*Actividad Precedente:* Programación.

*Actividad Siguiente:* usar el data warehouse

*Tiempo estimado:* -

*Frecuencia de ejecución:* -

*Objetivo:* ejecutar los programas desarrollados teniendo en cuenta:

- Frecuencia de ejecución.
- Purga de datos.
- Edad de los datos.
- Manejo de los múltiples niveles de granularidad.
- Refresco de los datos de ejemplo.

*Parámetros de Éxito:* el resultado de este paso es un data warehouse funcional y poblado que sirve a las necesidades de la comunidad DSS.



---

## **Sección 4**

### **Un ejemplo de diseño utilizando la Metodología**

---

La presente sección consta de dos capítulos. En el capítulo 6, Definición del Dominio, encontraremos la definición del sistema transaccional en el que está basado nuestro trabajo (Sistema Informático Asistencial), describiendo su funcionalidad, circuito administrativo general, descripción funcional de los módulos destacando el tipo de información de acuerdo a su importancia. En el capítulo 7, Desarrollo de la Metodología, desarrollaremos la metodología de diseño elegida para implementar el módulo de gestión propuesto, justificando dicha elección.

## Capítulo 6

# DEFINICIÓN DEL DOMINIO

## 6.1 Definición del S.I.A.

Es una herramienta destinada a entidades para la administración de la provisión de servicios de salud, la registración y seguimiento de denuncias de accidentes y el control de las liquidaciones de gastos relativos a la atención médica de la población beneficiada.

Actualmente el S.I.A. (Sistema Informático Asistencial) tiene un módulo de Gestión que permite realizar distintos tipos de estadísticas respecto a los distintos consumos que se realizan, ya sean éstos de prácticas o medicamentos. Además éstas estadísticas se realizan desde el punto de vista de los afiliados, prestadores, médicos, de una práctica o un medicamento en particular. Estos informes están basados en *datos operativos* que surgen de la actividad diaria de la organización. La *rigidez* de éste módulo no permite obtener información más que la provista por los pocos parámetros involucrados.

Para obtener información analítica para brindar nuevos planes de cobertura, programas, servicios, etc. los elementos necesarios están fuertemente relacionados con los consumos de prácticas médicas, ortesis, prótesis, odontología, etc. por parte de los afiliados o grupos de afiliados teniendo en cuenta las regiones geográficas, épocas del año, planes de cobertura, tipos de prestadores, etc.; y permitir las combinaciones necesarias entre éstos elementos. La rigidez del módulo de gestión del S.I.A. más el hecho de que estamos trabajando en el ambiente operativo (y su modelo) no hace posible satisfacer éstas necesidades.

## 6.2 Funcionalidad del S.I.A.

A continuación describiremos brevemente conceptos generales y la estructura de la información manejada por el S.I.A, dicha descripción es necesaria para entender los

fundamentos en los que nos basamos para la propuesta de nuestro diseño. Los entes o componentes que se tienen en cuenta son:

#### **Responsables de facturación.**

Son las entidades encargadas de presentar la facturación de los servicios prestados a los beneficiarios de la obra social, éstas pueden ser entidades propiamente dichas como colegios profesionales, clínicas, hospitales como así también profesionales directamente. En el caso de las entidades, éstas presentan la facturación de los prestadores agrupados en dicha entidad. Ejemplo: la clínica X presenta la facturación de los médicos prestadores de esa clínica, otro ejemplo el colegio de farmacéuticos presenta la facturación de las farmacias agrupadas en dicho colegio. En el caso de los prestadores no agrupados en ninguna organización éstos juegan el rol de prestador y responsable de facturación, claramente la estructura de la información que se maneja en este caso se vería jerarquizada de la siguiente manera:

Responsables de facturación

Prestadores

#### **Planes de cobertura**

Permite fijar las distintas coberturas que se brindarán a los afiliados por los distintos planes que determine la Obra Social.

Se denomina plan asistencial a la parte del plan donde se indican las distintas coberturas médico - asistenciales a las que va a tener acceso la persona en el momento de la afiliación. A través del plan se pueden fijar normas tales como topes de consumo, carencias, determinar autorizaciones, reintegros, manuales terapéuticos, etc.

Los planes de cobertura podrán estar agrupados en servicios que son los integrantes de la cobertura médica que agrupa prácticas con características similares, como por ejemplo, el servicio de consulta que puede incluir:

- Consulta en consultorio
- Consulta en domicilio
- Consulta con especialista
- Aquí la estructura de la información es:
  - Planes
  - Programas
  - Servicios

En los planes de afiliación se determinarán las normas que debe cumplir una persona para ser afiliado a la obra social.

### Afiliado

Los afiliados a la obra social son los beneficiarios de los servicios de salud prestados por la misma, ya sea titular o familiar a cargo, dicho afiliado goza de un plan de cobertura.

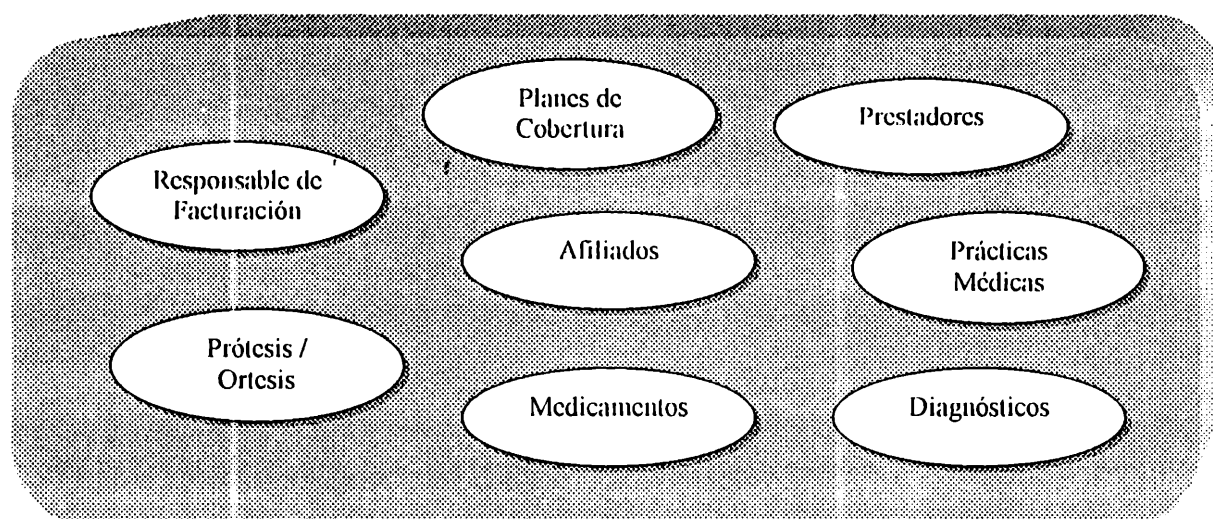
### Diagnóstico

Son los establecidos por la Organización Mundial de la Salud (O.M.S.), de manera que puedan asociar prácticas factibles de realizar, medicamentos a suministrar, etc.

### Prestaciones

Son los distintos servicios que brinda la obra social a sus afiliados que se pueden clasificar de la siguiente manera:

- Prácticas médicas ambulatorias / internación
- Medicamentos
- Prótesis / Ortesis / Optica



Componentes del S.I.A

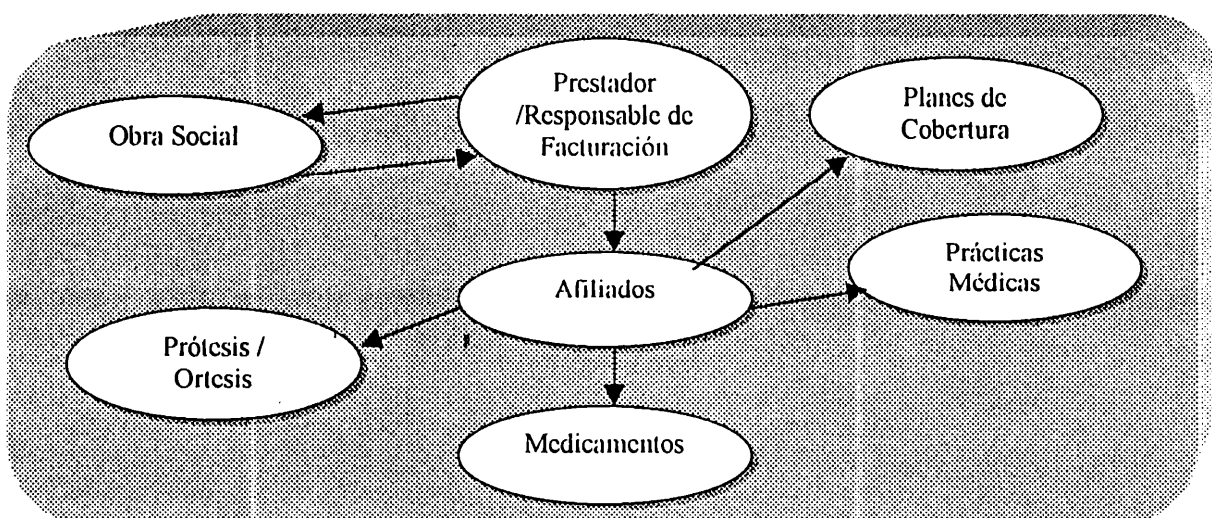
## 6.3 Circuito Administrativo General

Los conceptos e información mencionados previamente se relacionan de la siguiente manera:

El “prestador” brinda un servicio al afiliado según su plan de cobertura, dicho servicio se presenta a la obra social mediante la facturación correspondiente, siendo la cara visible ante la obra social el responsable de facturación de dicho prestador en

caso de que el prestador esté agrupado en alguna organización que actúa como responsable, sino el prestador es el responsable de facturación.

Luego la obra social se encarga de liquidar la facturación recibida realizando los controles correspondientes. Estos controles tienen en cuenta factores como el convenio realizado con el responsable de facturación, valores acordados, prácticas acordadas, topes, medicamentos, elementos de prótesis, ortesis, óptica, etc. De aquí surge la liquidación final para el responsable, débitos realizados por distintos motivos como por ejemplo que se haya facturado una práctica a un valor más alto que el acordado en el convenio. Este circuito produce información sobre los consumos de prácticas, medicamentos, prótesis, ortesis, óptica de la población beneficiada.



Relación entre los componentes

## 6.4 Descripción funcional de los módulos

A continuación presentamos la descripción funcional del Sistema Informático Asistencial, que será la base para el desarrollo de nuestro trabajo. Dicha especificación será limitada para tomar solo una parte del sistema en el desarrollo del diseño.

### Afiliaciones

Permite realizar nuevas afiliaciones en forma personal, actualizar y/o mantener la información referente a los afiliados tanto titulares como familiares a cargo.

Para implementar esta función es necesario contar con:

- El padrón general de afiliados totalmente saneado y convertido.

- El padrón de cada delegación a informatizar con sus procesos de actualización correspondientes.
- Todas las formas posibles por las cuales la información para afiliaciones ingresa a la Obra Social (medio físico, desde que entidades , etc.) con el fin de tener las especificaciones para desarrollar interfaces.

### **Planes de afiliación**

En los planes de afiliación se determinarán cuales son las normas que debe cumplir una persona para ser afiliado a la Obra Social .

### **Cuenta corriente de afiliados voluntarios**

En el caso de que la Obra Social cuente con la posibilidad de poder realizar afiliaciones de beneficiarios en forma voluntaria (no directos) el sistema contempla la posibilidad de poder mantener una cuenta corriente individual por afiliado voluntario, a los efectos de poder registrar pagos de cuotas, pago de coseguros, conceptos adicionales, reintegros. etc..

### **Emisión de cuotas (cargos)**

Este módulo permite generar en forma automática las cuotas de afiliación para todos aquellos afiliados en condición de voluntarios. Estas cuotas son imputadas directamente a la cuenta corriente de cada uno de los afiliados.

### **Cobro de cuotas de afiliados adherentes/conceptos adicionales**

En el caso de que la Obra Social admita afiliados adherentes mediante el pago de una cuota o que cuente con servicios adicionales al los cuales los afiliados acceden a través de un pago adicional, este módulo registra estas operaciones.

### **Autorizaciones**

El objetivo de este módulo es el de autorizar aquellos servicios que la Obra Social determine como norma antes de ser efectuada la prestación.

En ese acto se analizarán los consumos del afiliado en cuestión, carencias o topes y demás datos concernientes.

Este módulo contiene la siguiente apertura:

- Autorizaciones/Ventas de prácticas médicas ambulatorias.

Permite realizar la autorización/venta correspondiente a las prácticas ambulatorias que así lo requieran, controlando topes y validez de los componentes de la práctica solicitada, en forma simple y sencilla. Permite, además, realizar un control posterior contra la facturación de los prestadores. En el caso de la venta (también llamada coseguro) se realizará a su vez una valorización de las prácticas involucradas

permitiendo obtener el monto a abonar por el afiliado (normalmente un porcentaje del valor de la práctica).

- **Autorizaciones/Ventas de prácticas médicas en internación.**

Este módulo es similar al de prácticas ambulatorias pero siempre se refiere a una orden de internación previamente emitida por una determinación del departamento de auditoría médica.

- **Autorizaciones de medicamentos ambulatorios.**

Similar a los anteriores pero esta vez sobre los medicamentos suministrados por la red de farmacias a los afiliados en estado ambulatorio.

- **Emisión de órdenes de internación**

La emisión de órdenes de internación generará un documento con el cual el afiliado podrá solicitar la internación en el prestador de su elección, pudiendo la Obra Social mantener el control de estas situaciones registrando lugar de internación, diagnósticos, medico solicitante, días de internación, etc.

- **Provisión de prótesis y ortesis.**

Cuando sea necesario la provisión de una prótesis u Ortesis, el sistema cuenta con la posibilidad de generar una solicitud de autorización de prótesis y ortesis que, en relación directa con el módulo de presupuestos, permitirá al afiliado contar en el debido tiempo y forma con el elemento requerido.

### **Reintegros**

Permite operar con reintegros a realizar a los afiliados por prestaciones de servicios realizados en prestadores de la red o prestadores eventuales, según normativas de la Organización.

Permite, además, realizar controles sobre topes de reintegros y manejar distintas variables sobre la forma de realizar esta operación (valores a reintegrar).

La apertura de este módulo es la siguiente:

- Reintegros de prácticas médicas ambulatorias
- Reintegros de medicamentos ambulatorios
- Reintegros de prácticas en internación
- Reintegros de Prótesis/Ortesis/Óptica

### **Disposiciones/resoluciones (autorización por expediente).**

La emisión de una disposición o resolución es la forma de poder autorizar la utilización a los afiliados de aquellos servicios que les están inhibidos por las normas internas de la Organización.

Es posible la registración de disposiciones o resoluciones de:

- Prácticas médicas ambulatorias
- Prácticas médicas en internación
- Medicamentos ambulatorios
- Prótesis/Ortesis/Óptica

### **Subsidios**

Este módulo administra las asignaciones de distintos tipo de subsidios a los afiliados.

### **Nomencladores**

El módulo de nomencladores contiene toda la información administrativa, de auditoría médica y de valorización de todas las prácticas médicas cubiertas por la Obra Social.

### **Medicamentos**

El módulo de medicamentos contiene toda la información administrativa, de auditoría médica y de valorización de todos los medicamentos cubiertos por la Obra Social.

### **Nomenclador de prótesis/Ortesis/óptica**

Este modulo contiene toda información referente a las prótesis y Ortesis cubiertas por la Obra Social.

### **Diagnósticos de la Organización Mundial de la Salud (Protocolos)**

La forma que el sistema cuenta para poder protocolizar un diagnóstico es la de poder asociar a cada código de la O.M.S. las prácticas factibles de realizar, los medicamentos factibles de suministrar y, en el caso de ser un diagnóstico de internación, poder determinar la cantidad de días de estada.

### **Planes de cobertura**

Permite fijar las distintas coberturas que se brindarán a los afiliados por los distintos planes que determine la Obra Social.

Se denomina plan asistencial a la parte del plan donde se indican las distintas coberturas medico - asistenciales a las que va a tener acceso la persona en el momento de la afiliación. A través del plan se pueden fijar normas tales como topes de consumo, carencias, determinar autorizaciones , reintegros, manuales terapéuticos, etc.



Los planes de cobertura podrán estar agrupados en servicios que son los integrantes de la cobertura médica que agrupa prácticas con características similares, como por ejemplo, el servicio de consulta que puede incluir:

- Consulta en consultorio
- Consulta en domicilio
- Consulta con especialista

### **Prestadores**

En el módulo de prestadores se registrarán todos aquellos profesionales médicos y auxiliares de la medicina que prestan servicio a los afiliados de la Obra Social.

Como información adicional de los prestadores se podrán registrar las siguientes características

- Especialidades médicas
- Matrículas
- Acreditaciones
- Rubros y horarios de atención

### **Proveedores (entidades)**

En el módulo de proveedores se registrarán todas aquellas entidades que pueden suministrar prestaciones a los afiliados a la Obra Social en forma directa o indirecta.

### **Responsables de facturación (cuenta corriente proveedores).**

En el módulo de responsables de facturación se registrarán todas aquellas entidades que tienen un contrato o convenios con la Obra Social para el suministro de prestaciones médicas.

### **Convenios con prestadores asistenciales**

Se registrarán en este módulo todos los convenios que haya realizado la Obra Social con las entidades destinadas a prestar servicios médicos a los afiliados a la Obra Social.

Los diferentes tipos de convenios son:

- De facturación con prestadores médicos del tipo prestacionales
- De facturación con prestadores médicos por cápita

### **Control de facturación (solo contra presentación de documentación)**

La función de este módulo es la de controlar la facturación recibida de los prestadores generando las cartas de ajustes necesarias en aquellos casos en que lo presentado por el prestador no coincida con lo convenido en los contratos prestacionales.

Las áreas que abarca el control de facturación son las siguientes :

- Ambulatorio

- Internación
- Odontología
- Farmacia

### **Normas de facturación**

Se registrarán en este módulo todas las normas registradas en el nomenclador INOS a aplicar en el control de la facturación.

### **Seguimiento de documentación (mesa de entradas)**

Este módulo permite registrar el ingreso de la documentación a la Obra Social, asignarle un número de identificación, distribuirlo, desglosarlo, monitorearlo y archivarlo en forma física.

### **Proveedores (tratamiento, cuenta corriente).**

El presente modulo permite realizar el mantenimiento de datos del padrón de prestadores y proveedores del ente, teniendo en consideración detalles de categorías, actividades y situación fiscal de los mismos; así como también los datos referentes a la cuenta corriente de los mismos (deudas y pagos efectuados). Incluye distintas vistas de la información (consultas e informes) teniendo en cuenta datos de la deuda (tipo de moneda, vencimientos, etc.). Este módulo se interrelaciona con el modulo de convenios medico - asistenciales, siendo los prestadores asistenciales los responsables de facturación del mismo. También se contempla la generación de ordenes de pago, en forma individual o a medida que se auditan en forma *masiva*. Luego se procede a la liquidación - generalmente asociada con el sector de tesorería -, para, finalmente, realizar el pago

### **Caja Bancos.**

Su funcionalidad esta fundamentada en un dinámico manejo de las cajas sustentado en un modulo de seguridad apropiado de acuerdo a la personalización deseada por el usuario. Permite cierres parciales y definitivos, transferencias, depósitos y todas aquellas operaciones de ingresos y egresos.

El módulo de bancos permite el seguimiento de las operaciones relacionadas al área mencionada. Permite administrar cheques propios y de terceros. Muestra cotizaciones a partir de todas las monedas con que trabaje la institución. A su vez se complementa con una estructura de tablas primarias, que aseguran una parametrización a medida para el usuario

La liquidación de las ordenes de pago contempla pagos con más de un cheque, o parte en efectivo y parte en cheque, en diferentes monedas y permite parametrizar topes para optar entre una u otra forma de pago.

### **Contable.**

El concepto de contabilización en línea para la mayoría de las transacciones del sistema demuestra que se trata de un módulo ágil y dinámico.

Permite análisis intensivos y extensivos de la información contable, de la Casa Central y todas las delegaciones asociadas. Permite cierres parciales y definitivos. Existe la posibilidad de generar asientos tipo, teniendo en cuenta su repetición a lo largo de un periodo de tiempo determinado, disminuyendo de esta forma la operatoria de ingresos de asientos.

También permite la carga de asientos de ajustes o de cualquier otra índole, controlando que no se impute sobre cuentas afectadas a otros módulos - parametrizadas por el usuario -. Permite identificar el módulo que dio origen a cada asiento, y efectúa un control sobre ellos para evitar inconsistencias.

### **Ingresos**

Es un módulo que cuenta con la posibilidad de administrar cuentas corrientes de empresas y afiliados tanto directos como adherentes. Su funcionalidad está dada en la posibilidad de generar cualquier tipo de acuerdos a través de conceptos parametrizables, además de refinanciaciones de deudas. Posee actualización de deuda a una fecha determinada, permite recaudar por diversas vías (banco, DGI, caja).

Conceptualmente la cuenta corriente es algo más que un control de cobranzas. Su interrelación con los módulos anteriores asegura una confiabilidad y agilidad para el tratamiento de transacciones.

### **Compras**

Permite registrar las compras de suministros que hacen al funcionamiento general de la obra social -desde útiles de oficina, bienes de uso, instalaciones hasta inmuebles-, así como los más específicos como ser medicamentos, prótesis y ortésis.

Permite llevar un registro de cotizaciones de proveedores, así como los distintos métodos que utilice la organización para efectuarlas (concurso de precios, licitación, compra directa), parametrizando los topes y condiciones para optar entre los mencionados métodos. Luego, la apertura del expediente a través de la orden de compra y el control de entregas.

Una vez completado el circuito, con el ingreso de la factura, sigue el tratamiento de la documentación mencionada en el ítem de proveedores.

### **Activo Fijo**

Este módulo puede ser alimentado a través de los módulos de compras o de proveedores o de lo contrario, en forma directa.

Permite llevar un registro de los bienes de uso, muebles e inmuebles que posee la organización, y su vida útil para el cálculo de amortizaciones - que alimenta en forma directa al módulo contable - y para la generación de los cuadros que sobre estos bienes se exige presentar en el balance contable.

## 6.5 Tablas de clasificación de datos

A continuación presentamos las tablas de clasificación de los datos de los distintos módulos del sistema. Esta clasificación está basada en la importancia de los mismos para que cada módulo cumpla con su función. La información se clasifica en:

Crítica: información indispensable para el funcionamiento.

Funcional: necesaria pero no indispensable.

De gestión: conveniente para informes estadísticos, de gestión, etc..

PROVEEDORES		
INFORMACION	TIPO	OBSERVACIONES
Carga de Proveedores	CRITICA	
Carga de Actividades/ Cuentas por proveedor	CRITICA FUNCIONAL GESTION	<ul style="list-style-type: none"> <li>• Mínimamente una por categoría de retención</li> <li>• Imputación detallada del gasto por comprobante</li> <li>• Acota error de imputación</li> </ul>
Categorías de Retenciones/ Tablas de porcentajes aplicables	CRITICA FUNCIONAL	<ul style="list-style-type: none"> <li>• Fundamental para el correcto cálculo de retenciones de ganancias, ingresos brutos o las que correspondiere, de acuerdo al proveedor y la actividad</li> </ul>
Información Catastral por Proveedor	FUNCIONAL GESTION	<ul style="list-style-type: none"> <li>• Emisión de comunicaciones a los Proveedores</li> <li>• Informes de gestión sobre facturación por Pcia./Localidad/Zona</li> </ul>
Tipo de Proveedores	FUNCIONAL GESTION	<ul style="list-style-type: none"> <li>• Informes de gestión por tipo de proveedor</li> <li>• Emisión de comunicación para un determinado tipo de Proveedor</li> <li>• Asociación con la cuenta contable a imputar</li> </ul>
Identificación de Sucursales por Proveedor	CRITICA	<ul style="list-style-type: none"> <li>• Mínimamente se requiere identificar a una con su respectiva ubicación geográfica</li> </ul>
Identificación de servicios por proveedor	FUNCIONAL GESTION	<ul style="list-style-type: none"> <li>• Manejar ciertas restricciones por servicio en los convenios</li> <li>• Identificar servicios cubiertos por zona</li> </ul>
Tipos de Comprobantes	CRITICA	<ul style="list-style-type: none"> <li>• Identificar comprobantes posibles de ingresar y su signo para la cuenta corriente (Débito o Crédito)</li> </ul>

<b>CONTABLE</b>		
<b>INFORMACION</b>	<b>TIPO</b>	<b>OBSERVACIONES</b>
Carga del Plan de Cuentas	CRITICA	
Definición de datos accesorios (Nivel, mascara de impresión)	FUNCIONAL	<ul style="list-style-type: none"> <li>Permiten la posibilidad y/o la depuración de ciertos informes.</li> </ul>
Carga de Centros de Costos	FUNCIONAL GESTION	<ul style="list-style-type: none"> <li>Permite obtener información agrupada en formas convenientes de acuerdo a las unidades económicas que la empresa determine</li> </ul>
Carga de saldos por cuenta al momento de puesta en marcha	FUNCIONAL GESTION	<ul style="list-style-type: none"> <li>Permite obtener saldos reales por cuenta a una fecha</li> </ul>

<b>CAJA y BANCOS</b>		
<b>INFORMACION</b>	<b>TIPO</b>	<b>OBSERVACIONES</b>
Maestro de Bancos, sucursales	CRITICA FUNCIONAL	<ul style="list-style-type: none"> <li>Permite la realización de las distintas operaciones a realizar</li> <li>Permite identificar a los cheques en custodia por banco al que pertenecen.</li> <li>No es necesario identificar las sucursales donde la entidad en cuestión no posea cuentas abiertas.</li> </ul>
Chequeras (Carga del estado al momento de puesta en marcha)	FUNCIONAL	<ul style="list-style-type: none"> <li>Permite obtener información de los movimientos y control de stock de chequeras.</li> </ul>
Parámetros Generales	CRITICA	<ul style="list-style-type: none"> <li>Resulta fundamental para el vinculo de las posibles operaciones, cuentas a utilizar, etc.</li> </ul>
Cotizaciones	CRITICA	<ul style="list-style-type: none"> <li>Sirve para la realización de conversiones monetarias de manera exacta</li> </ul>
Valores del tesoro, saldos de cuentas bancarias	FUNCIONAL GESTION	<ul style="list-style-type: none"> <li>Util para la conformación de saldos de acuerdo a los valores. Carga manual UNICA a la puesta en marcha, luego la actualización es automática</li> </ul>
Cheques en custodia y en cartera al momento de puesta en marcha del sistema	FUNCIONAL, CRITICA	<ul style="list-style-type: none"> <li>Necesario para poder comenzar a operar en lo que va a ser los nuevos depósitos realizados y los pagos a proveedores.</li> </ul>

### INGRESOS

INFORMACION	TIPO	OBSERVACIONES
Maestros de Clientes, Afiliados, empresas	CRITICA	•
Conceptos de retención y porcentajes	CRITICA	•
Acuerdos, porcentajes	CRITICA	•
Tipos de Comprobantes	CRITICA	•

### AFILIADOS

INFORMACION	TIPO	OBSERVACIONES
Padrón de afiliados	CRITICA	
Normas generales de afiliación por plan	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permite controlar automáticamente las normas de afiliación de acuerdo a definiciones de la obra social. (Estado civil de hijos, edad de hijos, conflictos entre familiares a cargo como esposa y concubina, etc.)</li> </ul>
Definición y carga de Función /parentescos/clase de miembro	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permite identificar los roles de cada uno de los integrantes del grupo familiar</li> <li>• Permite valorizar afiliados voluntarios por su relación de parentesco con el titular</li> </ul>
Lugar de residencia del beneficiario y su grupo familiar como así también de familiares no convivientes.	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permite enviar comunicaciones por parte de la obra social al beneficiario</li> <li>• Permite obtener información de gestión por zona del afiliado</li> </ul>
Tipo de documentos	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permite identificar los documentos presentados por el afiliado, logrando de esta forma poder cargar los distintos documentos del mismo.</li> </ul>
Identificación del número de CUIL del beneficiario titular	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permitirá en el sistema de ingresos identificar los aportes recibidos por cada beneficiario y así podrá conformar su cuenta corriente</li> </ul>
Tablas Adicionales <ul style="list-style-type: none"> <li>■ Grupos Sanguíneo</li> <li>■ Nivel de estudio</li> <li>■ Nivel de discapacidad</li> </ul>	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permite clasificar al titular como así también a los integrantes del grupo familiar en características que los mismos posean</li> </ul>



PRESTADORES		
INFORMACION	TIPO	OBSERVACIONES
Carga de prestadores médicos	FUNCIONAL	<ul style="list-style-type: none"><li>Necesario para poder identificar a los prestadores efectores/prescriptores de las distintas instancias ; Autorizaciones, Ordenes de internación, etc.</li></ul>
Información Catastral por prestadores	FUNCIONAL GESTION	<ul style="list-style-type: none"><li>Emisión de comunicaciones a los prestadores</li></ul>
Determinación de la categoría del prestador	FUNCIONAL	<ul style="list-style-type: none"><li>A efectos del control de la facturación.</li></ul>
Matriculas por Prestador	FUNCIONAL	<ul style="list-style-type: none"><li>Necesario al momento de identificar rápidamente en la carga de información a través de la matricula informada en el sello y/o RP.</li></ul>
Especialidades por prestador	FUNCIONAL GESTION	<ul style="list-style-type: none"><li>A efectos de poder administrar particularidades en los convenios médicos</li><li>Información estadística por especialidad medica.</li></ul>
Lugar de Trabajo (Proveedor) y Horarios (Opcional)	FUNCIONAL	<ul style="list-style-type: none"><li>La identificación del lugar de atención y horarios para el armado de una cartilla medica y una rápida asistencia a los beneficiarios.</li><li>Poder realizar comunicaciones a los mismos.</li></ul>
Acreditaciones	GESTION	<ul style="list-style-type: none"><li>Contar con la información suficiente para la contratación de los profesionales.</li></ul>

<b>NOMENCLADOR</b>		
<b>INFORMACION</b>	<b>TIPO</b>	<b>OBSERVACIONES</b>
Carga de los nomencladores con los que opera la institución.	CRITICA	
Identificación de Tipo de practicas	FUNCIONAL GESTION	<ul style="list-style-type: none"> <li>• Permite dar el tratamiento particular por el tipo de practica.</li> <li>• Permite clasificar las mismas a efectos de información estadística</li> </ul>
Textos de Llamado de atención	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permite notificar al operador al momento de una autorización/Reintegro a través de una alarma de ciertas particularidades como por ejemplo documentación a exigir o pasos a seguir para el caso en cuestión.</li> </ul>
Valores del nomenclador	CRITICA	
Practicas por Estado	CRITICA	
Practicas por especialidad	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permite identificar al momento de la autorización, control y de forma posterior posibles desvíos en la prescripción/realización de prácticas médicas según especialidad del profesional.</li> </ul>
Practicas por Sexo	FUNCIONAL	<ul style="list-style-type: none"> <li>• Necesario para disparar ciertas alarmas ante situaciones de realización de practica a sexo incompatible con esta.</li> </ul>
Grupo de Practicas	GESTION	<ul style="list-style-type: none"> <li>• Permite obtener información de gestión por grupos de practicas</li> </ul>
Incompatibilidades por edad	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permite realizar controles por edad del afiliado y prestaciones solicitadas.</li> </ul>
Incompatibilidades entre practicas	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permite realizar controles de incompatibilidad sobre practicas solicitadas al beneficiario en un tiempo determinado.</li> </ul>
Precexistencia	FUNCIONAL	<ul style="list-style-type: none"> <li>• Necesario para realizar controles de precexistencia sobre practicas solicitadas.</li> </ul>



<b>MEDICAMENTOS</b>		
INFORMACION	TIPO	OBSERVACIONES
Carga de los Medicamentos	CRITICA	
Actualización de precios a través de alguna de las empresas del mercado	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permite manejar menor tiempo de actualización que cualquier método no automatizado.</li> <li>• Da seguridad y mayor respaldo</li> </ul>
Clasificación de Medicamentos/ Apertura de información por medicamentos	GESTION	<ul style="list-style-type: none"> <li>• Esta apertura de información permite mayor variedad y mas detalle en cuanto a información sobre consumo de medicamento, tasas de uso, etc.</li> </ul>

<b>VENTA DE ORDENES</b>		
INFORMACION	TIPO	OBSERVACIONES
Tipo y valor de las ordenes	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permite diferenciar los distintos tipos de ordenes vendidas.</li> <li>• Permite valorizar la venta de las mismas a efectos de control, rendición e información de gestión sobre las mismas.</li> </ul>

<b>DIAGNOSTICOS</b>		
INFORMACION	TIPO	OBSERVACIONES
Carga de los diagnósticos	CRITICA	
Estado de los diagnósticos	FUNCIONAL GESTION	<ul style="list-style-type: none"> <li>• Apunta a permitir realizar un control sobre las patologías atendidas en los distintos estados.</li> <li>• Permite obtener información sobre patologías por los distintos estados.</li> </ul>
Armado de protocolo por patología o diagnostico	FUNCIONAL GESTION	<ul style="list-style-type: none"> <li>• Fuente de consulta de profesionales propios (Auditores) y de la red.</li> <li>• Permite que se logren realizar controles sobre los protocolos pautados por la Obra Social.</li> <li>• Facilita la visualización por prestador o a nivel general de desvíos en los consumos efectuados.</li> </ul>

<b>CONVENIOS</b>		
<b>INFORMACION</b>	<b>TIPO</b>	<b>OBSERVACIONES</b>
Carga Convenios	CRITICA	
Carga de detalle de contenido de los convenios como practicas y/o medicamentos cubiertos	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permite el control de la facturación</li> <li>• Permite la valorización de las autorizaciones a valores pactados con el prestador</li> <li>• Información de servicios brindados por zona geográfica a través de los responsables de los convenios. (Los mismos deben tener información catastral)</li> <li>• Control sobre practicas fuera de capita.</li> </ul>
Area de cobertura de convenios de capita	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permite Ejercer el control sobre capitas por afiliados en transito. (En caso que dicho control recaiga sobre la obra social)</li> </ul>

<b>PLANES DE COBERTURA</b>		
<b>INFORMACION</b>	<b>TIPO</b>	<b>OBSERVACIONES</b>
Carga de Planes de cobertura	CRITICA	
Manual terapéutico o características de cobertura farmacéutica	FUNCIONAL GESTION :	<ul style="list-style-type: none"> <li>• Control sobre descuentos en prestaciones farmacéuticas</li> <li>• Control sobre Topes de uso en prestaciones farmacéuticas</li> <li>• Determinación de la posibilidad de realización de reintegros</li> <li>• Topes por reintegro</li> </ul>
Definición de subsidios	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permite liquidar distintos tipos de subsidios</li> <li>• Control sobre tiempos y cantidad de subsidios entregados por beneficiario</li> </ul>
Servicios Médicos	FUNCIONAL	<ul style="list-style-type: none"> <li>• Permite separar las practicas medicas cubiertas por tipo de servicio de acuerdo a su naturaleza y características de cobertura.</li> <li>• Posibilita controles específicos por servicio.</li> </ul>
Características de cobertura Medica		<ul style="list-style-type: none"> <li>• Control sobre el cobro de coseguos</li> <li>• Control sobre tope de uso</li> <li>• Determinación de carencias</li> <li>• Determinación de la posibilidad de la realización de reintegros</li> <li>• Topes por reintegro/autorización</li> </ul>

<b>PRÓTESIS Y ORTESIS</b>		
<b>INFORMACION</b>	<b>TIPO</b>	<b>OBSERVACIONES</b>
Carga de nomenclador de prótesis	CRITICA	

<b>AUTORIZACION REINTEGROS</b>		
<b>INFORMACION</b>	<b>TIPO</b>	<b>OBSERVACIONES</b>
Parámetros generales de ventas (Nivel auditoria del sistema, Cantidad de practicas por comprobante, Control de servicios por impresión, tipos de control, etc. )	FUNCIONAL	<ul style="list-style-type: none"> <li>Determinar pautas para control en emisión de autorización/reintegros de practicas/medicamentos.</li> </ul>
Revisión, Codificación y clasificación de los motivos de Error en Autorizaciones/reintegros	FUNCIONAL	<ul style="list-style-type: none"> <li>Revisar la lista de errores suministrada por el sistema</li> <li>Codificar nuevos motivos de error</li> <li>Clasificar todos los errores según el criterio de la organización.</li> </ul>
Marcas de auditoria	FUNCIONAL	<ul style="list-style-type: none"> <li>Revisar la lista de marcas suministrada por el sistema</li> </ul>
Origen de Autorización	FUNCIONAL	<ul style="list-style-type: none"> <li>Revisar la lista de orígenes suministrada por el sistema</li> </ul>
Nivel de Autorización	CRITICO	<ul style="list-style-type: none"> <li>Revisar/confeccionar la lista de niveles de autorización</li> </ul>
Nivel de Auditoria	CRITICO	<ul style="list-style-type: none"> <li>Revisar/confeccionar la lista de niveles de auditoria</li> </ul>
Niveles de autorización/auditoria por usuario	CRITICO	<ul style="list-style-type: none"> <li>Determinar los niveles de acceso de los diferentes usuarios para realizar auditorias sobre practicas y/o medicamentos</li> </ul>

<b>FACTURACION MEDICA</b>		
INFORMACION	TIPO	OBSERVACIONES
Normas Generales de Facturación (Redondeos, Vencimiento de Bonos, Limite de líneas por Bono, Vencimientos de presentación, etc.)	FUNCIONAL	<ul style="list-style-type: none"> <li>Involucra parte de los controles que realiza el sistema. Sin esta información se resta una importante cuota del control que realiza el sistema.</li> </ul>
Codificación y clasificación de códigos de error	FUNCIONAL	<ul style="list-style-type: none"> <li>Crítico a nivel de determinar de acuerdo a las normas de control, débitos totales, parciales y advertencias sobre la facturación del prestador.</li> </ul>

<b>FACTURACION FARMACIA</b>		
INFORMACION	TIPO	OBSERVACIONES
Normas Generales de Facturación (Redondeos, Vencimiento de recetas, Limite de líneas por receta, Vencimientos de presentación, etc.)	FUNCIONAL	<ul style="list-style-type: none"> <li>Involucra parte de los controles que realiza el sistema. Sin esta información se resta una importante cuota del control que realiza el sistema.</li> </ul>
Codificación y clasificación de códigos de error	FUNCIONAL	<ul style="list-style-type: none"> <li>Crítico a nivel de determinar de acuerdo a las normas de control, débitos totales, parciales y advertencias sobre la facturación del prestador.</li> </ul>

## Capítulo 7

# DESARROLLO DE LA METODOLOGÍA

## 7.1 Pasos del desarrollo

### Paso 1: Análisis del modelo de datos.

- Identificar el / las áreas objetivo.

Para poder identificar el área o áreas objetivos de nuestro desarrollo debemos tener presente que la persona que realiza el análisis de la información debe tener todos los elementos necesarios que le permitan formar una base de conocimiento sólida del tema sobre el que tendrá que tomar decisiones futuras y obviamente que respalden dicha toma de decisión. En consecuencia el gran interrogante ante el que nos encontramos al intentar identificar el área objetivo es saber exactamente cuál va a ser la información que el analista tendrá que evaluar teniendo en cuenta que el propósito de dicho análisis será decidir futuras políticas de captación de adherentes a la Obra Social, para esto deberá implementar la creación o modificación de los servicios disponibles para los diferentes planes de cobertura para los afiliados actuales con el fin de no perder su afiliación, creación de nuevos planes de cobertura con servicios que apunten a la satisfacción de los potenciales o futuros afiliados.

Luego de evaluar el objetivo que persigue la persona que analizará y de la información con la que cuenta actualmente en sus sistemas transaccionales, en sus bases de datos operacionales y que pueden ser explotadas, llegamos a la conclusión que para lograr tal objetivo el foco que se debe atacar principalmente es el área constituido por los Consumos de los Afiliados ya que analizándolos desde distintas perspectivas, (que luego serán explicadas), podemos ver tendencias futuras, estadísticas de diversos tipos que permitan optimizar los servicios brindados desde el punto de vista económico equilibrando la relación ingreso/egreso .

Por todo lo expuesto identificamos el área objetivo *Consumos de Afiliados* para este desarrollo particular. Desde el punto de vista del análisis de la información una posible consulta sería la siguiente:

Consumos de prácticas quirúrgicas asociadas a un diagnóstico de afiliados con un ingreso promedio mayor a \$1000, con grupo familiar de 4 personas a cargo que se atendieron en una determinada entidad durante una cierta época del año.

- Definir los límites del modelo.

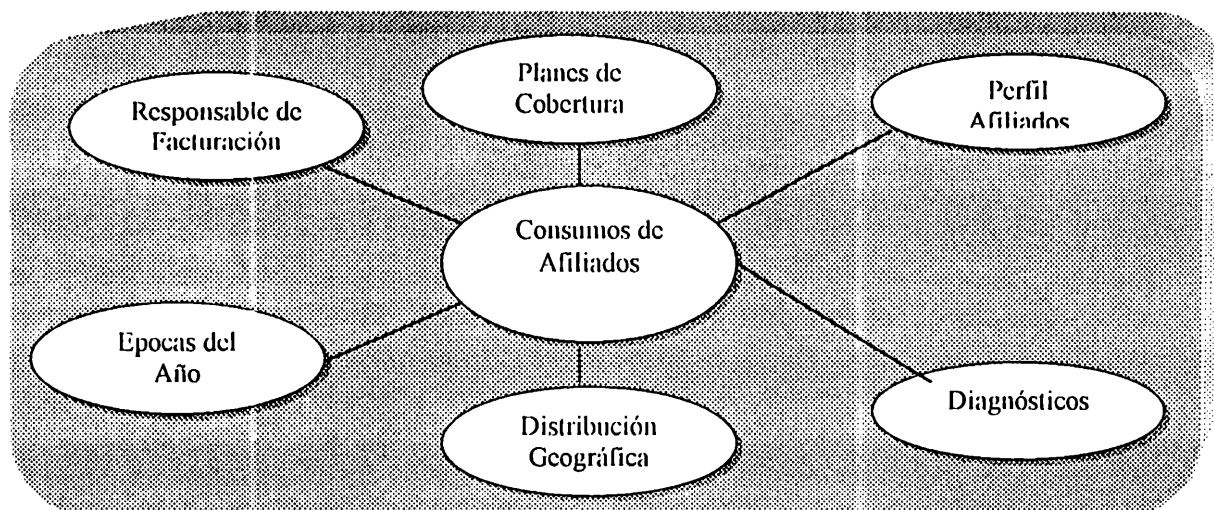
Teniendo en cuenta que el objetivo de este diseño es brindar una base sólida para el análisis de la información producida por el sistema con propósito de dar soporte a la toma de decisiones, encontramos de vital importancia que dicho análisis pueda realizarse sobre la población beneficiada, en particular, por franjas o segmentos de la misma, lo que determinan la necesidad de contar con *Perfiles de Afiliado*, ya que interesa la información de consumos relacionados a grupos con diversas características y no a beneficiarios individuales. También encontramos la necesidad de analizar dichos consumos por *distribución geográfica*, *Responsable de Facturación*, *Diagnósticos* y por *épocas del año*.

- Separar los datos primitivos de los derivados.

Los datos primitivos, pueden ser: descriptivos, por ejemplo *descripcion plan, programa, servicio*, etc.; o calculados en el operacional, por ejemplo *cantidad personas atendidas, total practicas*, etc.

Los datos derivados son datos calculados de los primitivos, por ejemplo *promedio practicas, totales*, etc..

En el Paso 7 se describirán cada uno de los datos indicando si es primitivo o derivado.



## **Paso 2: Análisis breadbox.**

- Encontrar la granularidad de los datos:

Como finalidad de este análisis tendríamos que obtener una granularidad de datos aconsejable para que el analista DSS pueda realizar sus consultas de forma rápida y eficiente, por lo tanto, debemos evaluar el tipo de consolidación de datos para cada una de las tablas del modelo relacional que funcionan como base de nuestro warehouse.

En el sistema operacional, encontramos que la actualización de datos se produce en forma diaria, generando así una gran cantidad de información. Teniendo en cuenta los límites del modelo enunciados en el Paso 1, utilizar este tipo de granularidad (diaria) no sería aconsejable, pues obtener la respuesta a una consulta involucraría el procesamiento de gran cantidad de registros para obtener una sumarización de datos más acorde a las necesidades del analista. Para evitar que la sumarización de los datos se realice cada vez que el analista requiera de los mismos, proponemos la consolidación mensual de los datos como fuente de nuestro modelo.

## **Paso 3: Distribución Técnica.**

Utilizaremos un DBMS multidimensional, el cual permite realizar consultas de grandes volúmenes de información con buenos tiempos de respuesta ya que posee una organización más flexible de datos y mayor dinámica de exploración entre sumarizaciones y detalles. Este tipo de base de datos está orientado tanto a usuarios finales como analistas DSS.

## **Paso 4: Preparación del entorno Técnico.**

Para realizar debidamente este paso necesitamos analizar la organización desde el punto de vista informático haciendo referencia al tipo de equipo con el que cuenta para la implementación de nuestro warehouse. Como primer punto de este análisis debemos identificar la arquitectura existente, tomando como base el servidor, el sistema operativo y las PC's clientes, estudiar la red, prestando atención sobre la velocidad de transferencia, pues ya sabemos que las consultas que el usuario realizará sobre el warehouse generarán un tráfico de red importante.

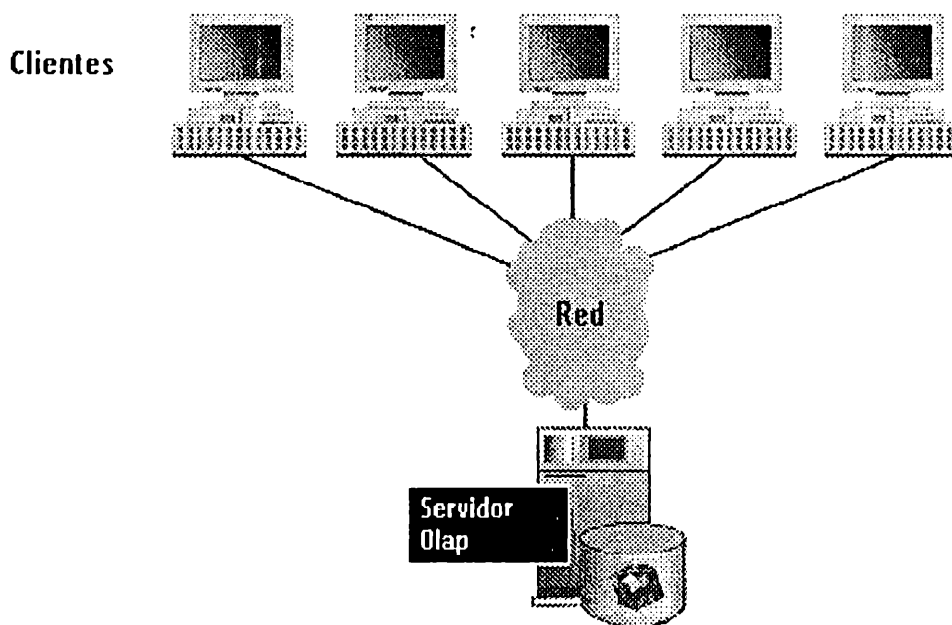
Otro punto importante en el análisis del entorno técnico es calcular la cantidad de almacenamiento necesario para cada una de las tablas de nuestro modelo (tablas de

dimensión y fact tables). Este cálculo se puede realizar teniendo en cuenta los siguientes parámetros:

- El horizonte de la tabla: determina la cantidad de unidades de tiempo que la misma mantendrá on line en la base de datos, un ejemplo de esta unidad podría ser año.
- Cantidad estimada de filas para una unidad de tiempo, por ejemplo, 30000 filas por año.
- Cantidad de bytes por fila.
- El espacio total se obtiene multiplicando cada uno de los parámetros mencionados anteriormente.

Como conclusión a este análisis planteamos dos soluciones posibles:

1. Utilizar un servidor OLAP para atender a todos los clientes y una base de datos multidimensional como motor. Gráficamente:

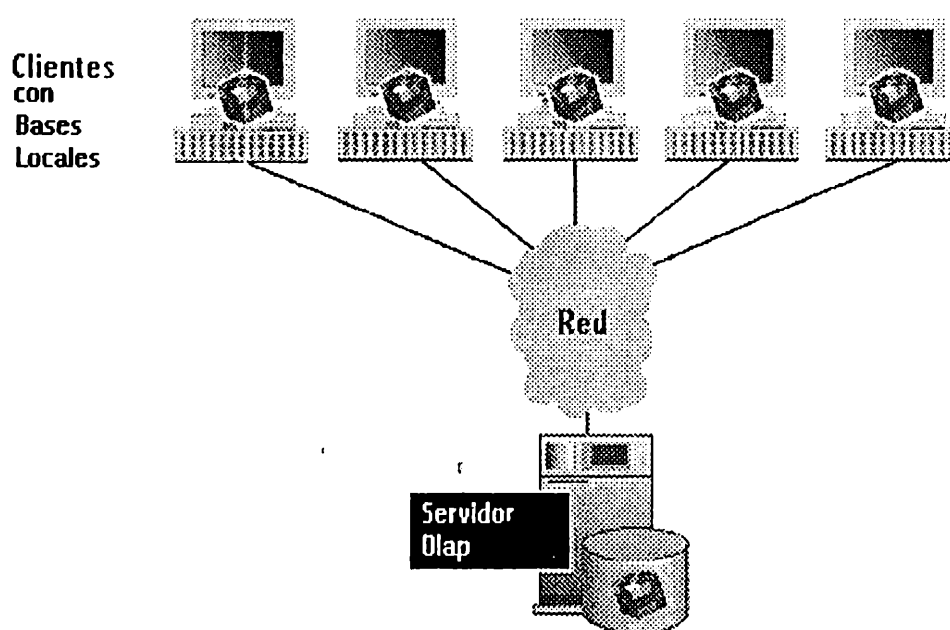


Para optimizar los tiempos de respuesta sería aconsejable contar con un servidor “potente” en términos de procesadores, memoria y dispositivos de almacenamiento eficientes, pues la sobrecarga de trabajo será muy alta. También tendremos que tener



en cuenta la arquitectura de red utilizada que funcionará como nexo entre la PC cliente y la base de datos multidimensional.

2. Para mejorar la performance de la solución anterior agregamos bases de datos locales multidimensionales independientes para cada PC cliente. Disminuyendo así el tráfico de red y la sobrecarga del servidor OLAP, pues cada una de las PC's funciona como servidor independiente. Como contrapartida existe la replicación de datos y la necesidad de contar con un programa que alimente las bases de datos locales. Gráficamente:



La elección del entorno técnico dependerá de la cantidad de analistas DSS que hagan uso del sistema, ya que la segunda opción es apropiada cuando la cantidad de usuarios es reducida, mientras que la primera opción es aplicable en cualquier caso.

### **Paso 5: Análisis del área objetivo.**

En este punto debemos plantear la primer área objetivo a ser poblada, que debe ser lo suficientemente grande e importante para ser significativa y lo suficientemente chica para ser implementada rápidamente.

Por lo tanto, de las entrevistas realizadas con el Analista se decide que en una primera etapa, la información crítica indispensable para un mínimo análisis es contar con datos referentes a:

- Responsables de facturación con sus respectivos prestadores,
- Prácticas Médicas válidas
- Perfiles de Afiliados
- Planes de Cobertura.
- Consumos que involucren a los anteriores.

Ya que éstos componentes son los más importantes de una obra social y con los cuales es posible hacer proyecciones, comparaciones de consumos que ayuden a la toma de decisiones, como por ejemplo en la creación y/o modificación de planes de cobertura.

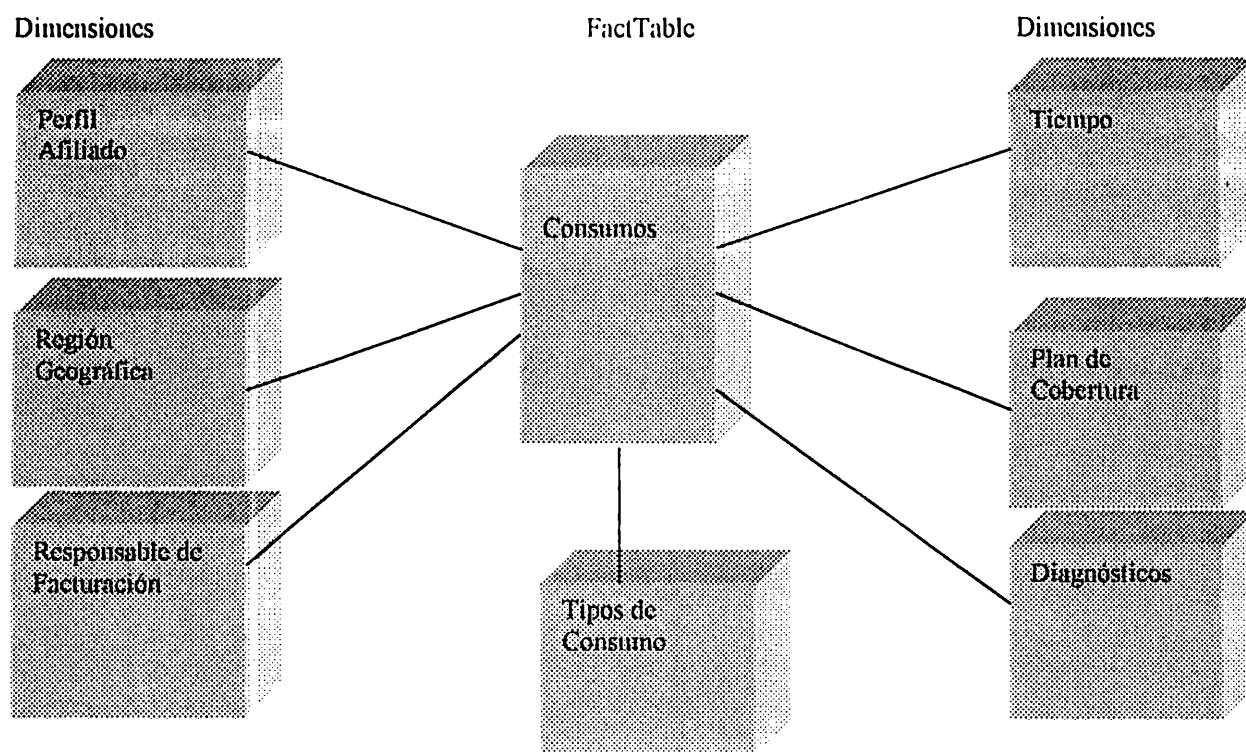
## **Paso 6, 7: Diseño del Data Warehouse – Análisis del sistema fuente.**

Como mencionamos en el Paso 1, nuestro trabajo estará enfocado en el área constituida por los Consumos de los Afiliados de la Obra Social. Las perspectivas desde las cuales podemos analizarlos para satisfacer los objetivos mencionados son los Planes de Cobertura que ofrece, los Diagnósticos y Prácticas Médicas establecidos por la Organización Mundial de la Salud, los Perfiles de Afiliados (mencionados en la definición de los límites del modelo), Ubicación Geográfica, Responsable de Facturación y Epocas del Año.

Estas entidades – consumos, afiliados, planes de cobertura, diagnósticos, prácticas médicas – son los principales componentes de la Obra Social y los que deben ser analizados en conjunto para tomar decisiones apropiadas. Debemos notar que estas entidades se poblarán en forma dispar, por ejemplo, Planes, Diagnósticos, Prácticas Médicas y Afiliados se poblarán con pocos datos, mientras que la entidad Consumos será densamente poblada. Debido a esta diferencia en volumen de datos que poblará a esta entidad es que necesita un tratamiento de diseño distinto.

El modelo de datos necesario para manejar gran volumen de información es el modelo dimensional o Star Join. Este modelo está compuesto por un conjunto de tablas relacionadas de la siguiente manera: una tabla central, llamada fact table (tabla de hechos) en donde se almacenan los datos numéricos, aditivos, esta es la tabla que será densamente poblada; y una serie de tablas, llamadas tablas de dimensión que almacenan datos descriptivos, éstas se unen, mediante claves foráneas, a la fact table formando la estrella.

Utilizamos este modelo para nuestro desarrollo. Como primer análisis obtenemos el siguiente esquema estrella:

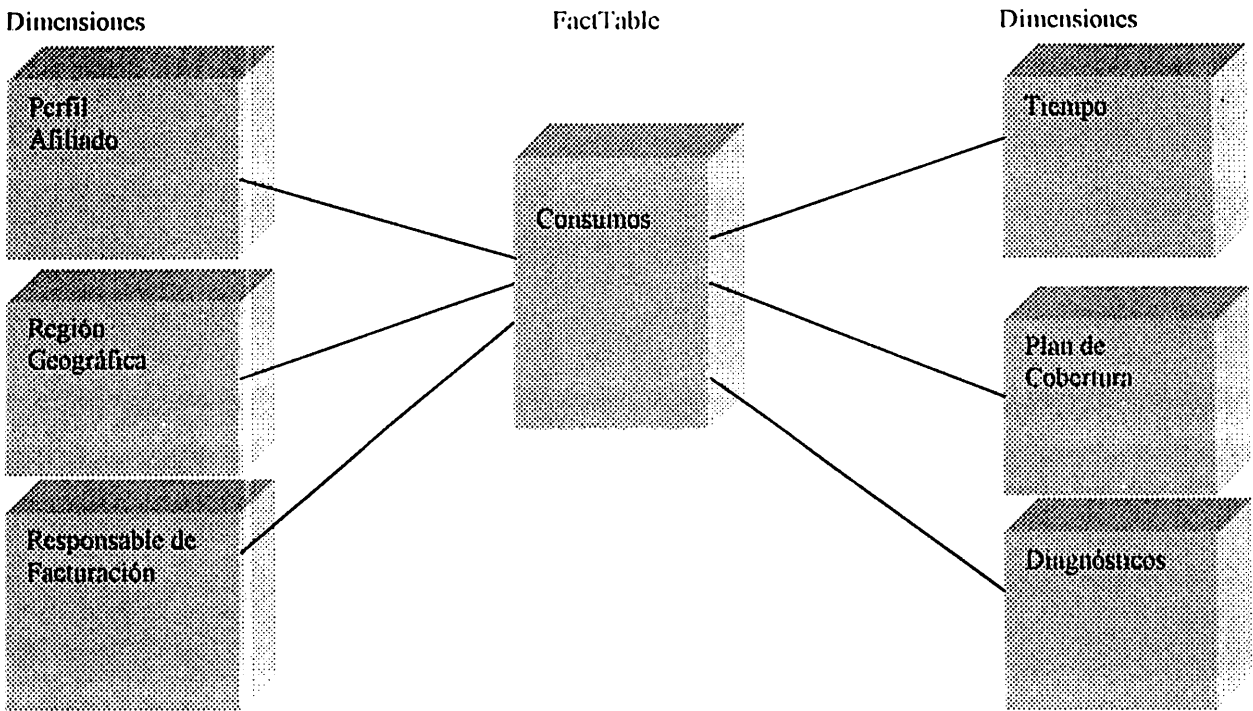


Donde la fact table de *Consumos* agrupa los consumos (cantidades, montos facturados, liquidados, promedios, etc.) realizados por los afiliados de la obra social en una época del año, en un lugar geográfico ante un determinado diagnóstico, en una entidad prestadora y según su plan de cobertura.

La dimensión *Perfil de Afiliado* agrupa a los beneficiarios de la obra social según ciertos criterios como ser edad, estado civil, ocupación, etc..

La tabla *Tipos de Consumo* se refiere a consumos de Prácticas Médicas, Elementos de Prótesis, Ortesis u Óptica o Medicamentos, éstos tienen características muy diferentes (datos heterogéneos), por lo que no pueden ser agrupados en una sola tabla. En un data warehouse donde una dimensión debe describir una gran cantidad de ítems heterogéneos, la técnica recomendada es crear una fact table y una dimensión centrales para permitir las consultas a través de los tipos dispares, y crear una fact table y una dimensión a medida para consultar cada tipo individual. Aplicamos esta solución definiendo un esquema central en donde están los consumos sin diferenciar el tipo; y tres esquemas a medida, que describen los consumos de acuerdo a su tipo. Es decir, nos quedan cuatro esquemas, en todos ellos se mantienen las dimensiones Perfil\_afiliado, Region\_Geográfica, Responsable\_Facturación, Tiempo, Plan\_cobertura y Diagnóstico.

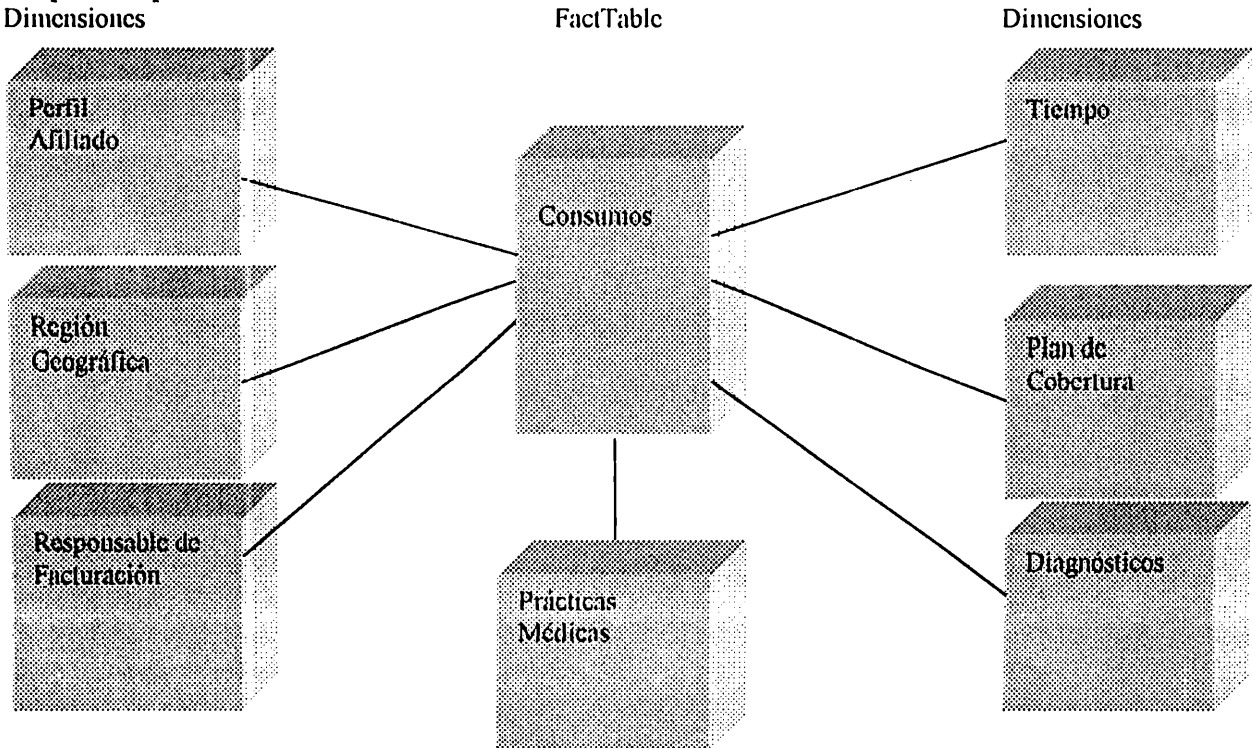
Esquema Central



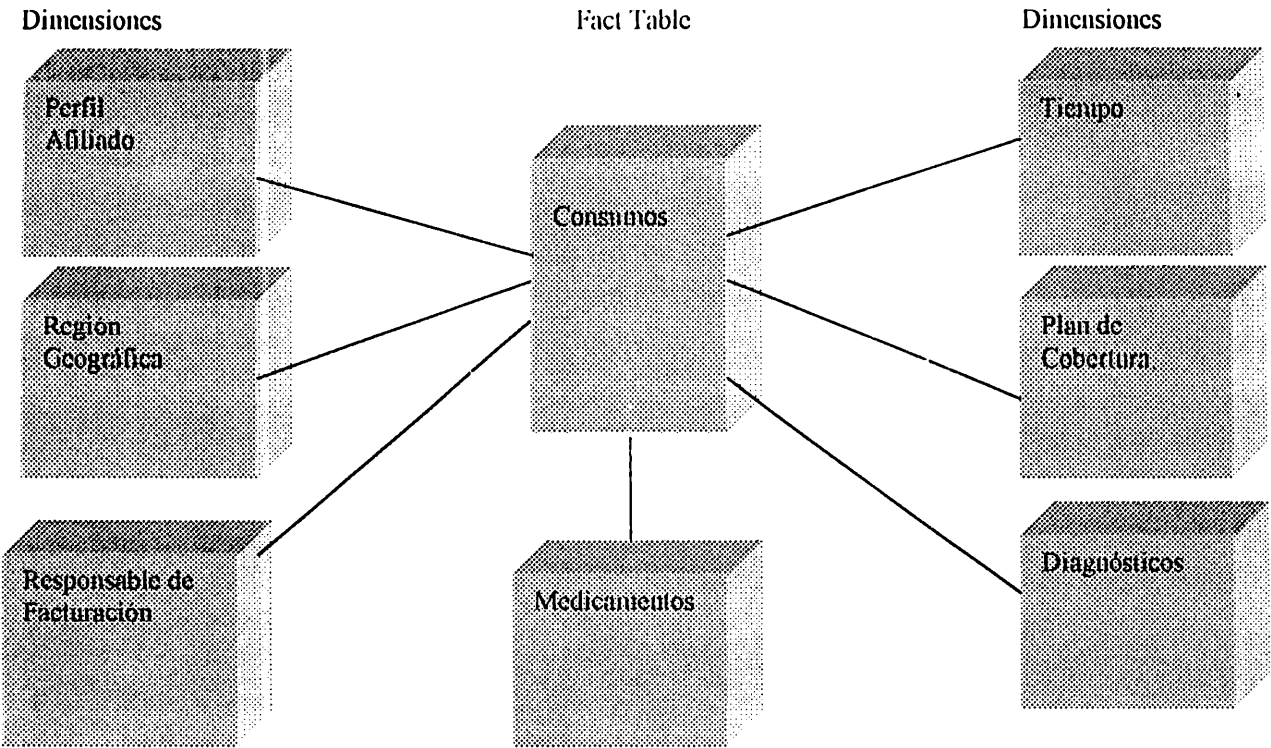
Esquemas a medida, según el tipo de consumo:

Los esquemas a medida tienen, además de las dimensiones comunes, una dimensión que describe el tipo de consumo. Estos esquemas son: Esquema para Prácticas Médicas, Esquema para Medicamentos y Esquema para Prótesis, Ortesis y Optica.

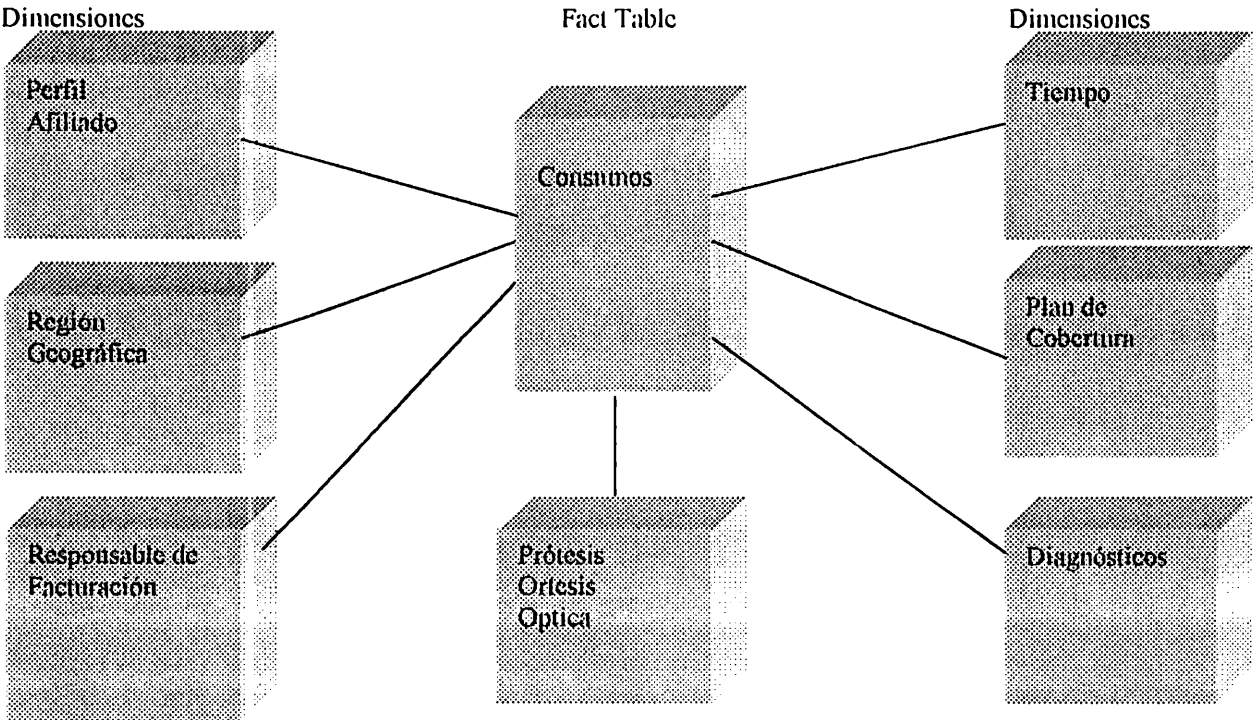
Esquema para Prácticas Médicas



Esquema para Medicamentos:



Esquema para elementos de Prótesis, Ortesis y Óptica



La descripción de las tablas de dimensión y fact tables de cada uno de estos esquemas se encuentra en el Anexo. Para cada una de ellas se indica:

- *Tipo de Tabla*: Dimension ó Fact Table
- *Nombre del campo o atributo*
- *Tipo* : Carácter, Numérico, Fecha
- *Origen del dato*: Tabla y atributo del modelo operacional
- *Derivado / Primitivo* : Clasificación de los atributos
- *Descripción* : Un breve comentario del atributo.

## Paso 8-Especificaciones

Una vez definida la interfaz entre los ambientes operacional y de DSS, el próximo paso es formalizarla en términos de especificaciones de programas. Estos son algunos de los puntos importantes:

- Cómo sé que datos operacionales buscar ?
  - Están los datos operacionales marcados con fecha y hora?
  - Hay un archivo delta ?
  - Hay registros de auditoría que puedan ser usados ?
  - Se pueden cambiar el código fuente existente y la estructura de datos para crear un archivo delta ?
- Cómo almaceno el resultado, una vez buscado?
  - Están los datos de DSS prealocados, preformateados?
  - Se adicionan los datos?
  - Se reemplazan los datos?
  - Se realizan actualizaciones en el ambiente del DSS.

Básicamente la mayoría de los archivos con datos relevantes para la extracción desde el ambiente operacional hacia el ambiente del Data Warehouse poseen fecha y hora de actualización, esto permitirá una búsqueda de datos más eficiente ya que aquellos datos que no hayan sido modificados o actualizados en el lapso de tiempo que se está analizando no serán procesados.

Como criterio general vamos a utilizar el concepto de agregación de datos al ambiente del Warehouse pudiendo mantener de esta manera la historia de datos relevantes para analizar sus cambios o evolución.

## Paso 9-Programación

Este paso incluye todas las actividades standard de programación, tales como:

- Desarrollo de pseudocódigo
- Codificación
- Compilación
- Testing

Cabe aclarar que para el caso particular de este Trabajo de Grado que incluye hasta la etapa de diseño del Data Warehouse los pasos previamente mencionados (Paso 8 y Paso 9) no tiene una implementación específica, tal realización podría estar contemplada en una etapa posterior de este proyecto o podría ser en sí misma un Trabajo de Grado tomando como base el actual.

Como el objetivo de nuestro trabajo es realizar el diseño y no la implementación, en este punto sólo hacemos mención a las características que debería brindar el producto elegido. Examinaremos dos posibles productos:

- Herramienta Star Tracker.
- Suite Oracle Express.

### StarTracker

Ha sido desarrollada como una herramienta para responder a preguntas de negocios o comerciales de la manera más útil para éstos. Mientras que la mayoría de las herramientas de consultas actuales pueden hacer un buen trabajo en la formulación de reportes y consultas simples, a menudo fallan en su capacidad de producir métricas y proporciones comerciales sofisticadas y significativas que se pueden extraer de las bases de datos grandes y complejas.

Las grandes cantidades de datos transaccionales desde puntos de ventas y sistemas bancarios, combinados con la alta capacidad y los bajos costos de los servidores de bases de datos relacionales introducen una nueva fuente de información que pide técnicas analíticas nuevas y altamente sofisticadas. *StarTracker* provee estas técnicas y está específicamente diseñada para ser usada en el análisis de bases de datos extremadamente grandes.

Las principales características que destacan a *StarTracker* son las siguientes.

- Soporte explícito para el modelo dimensional de un negocio a través de SQL.

- Estrategia de multiconsultas(multiqueries) para ensamblar reportes complejos que involucren comparaciones, múltiples granularidad de tiempo, múltiples agrupamientos de productos y múltiples agrupamientos de venta y geográficos.
- Creación del conjunto de consultas mínimo óptimo para reportes multiconsultas.
- Soporte de joins externos para mezclar conjuntos de respuesta separados.
- Campos protegidos en el visualizador (browser) para inhibir la búsqueda involuntaria de listas muy largas.
- Grupos de restricción públicos y privados definidos por el usuario.
- Una biblioteca de *funciones de comparación* entre grupos de restricciones.
- Una biblioteca de *funciones secuenciales* imposibles de programar en SQL, incluyendo tertiles, cuartiles, promedios, etc.
- Extensiones especiales a los operadores de agregación de SQL para *sumas (sums)* y *cuentas (counts)* *promedio de período* que soportan cálculos de inventarios y balances de períodos promedio como también comparaciones de medidas aditivas a través de intervalos de tiempo disímiles.
- *Filas de corte* con el manejo correcto de todas las medidas simples y compuestas.
- Resaltado Rojo y Verde para *valores excepcionales* en las columnas del reporte usando cinco criterios.
- Capacidad de perforar, profundizar (drill-down) que asigna un blanco separado de perforación o profundización separado para cada columna del reporte.
- Modos de columna completa y profundización(drill-down) de celda seleccionada.
- Creación de *grupos de estudio* que son registros de resultados de comportamiento difíciles de expresar en términos de restricciones sobre atributos.
- Creación de *grupos de estudio derivados* vía unión, intersección, y diferencia de conjuntos.
- Copia de reporte de celda seleccionada y de reporte completo a través del portapapeles a otras aplicaciones.



## Oracle Express

La familia de productos *Oracle Express* ofrece las funciones típicas de consulta, análisis e informes del tipo OLAP. En esta familia encontramos los siguientes productos:

*Oracle Express Server*: Es el componente central de esta familia, es el motor de la base de datos. Este motor soporta funcionalidad relacional y multidimensionalidad OLAP.

*Oracle Express Administrator*: Es una herramienta gráfica que se utiliza para la administración de la base *Express*, permitiendo creación de los componentes de la base (variables, dimensiones, procedimientos) como así también realizar la importación y exportación de datos.

*Oracle Express Analyzer y Express Object*: Son productos que permiten el desarrollo de una aplicación, donde el entorno de desarrollo con *Express Analyzer* está orientado a los usuarios finales (analistas DSS), mientras que el *Express Object* es una herramienta programación pues cuenta con un lenguaje de procedural que se utiliza para el desarrollo de una aplicación.

*Oracle Express Web Publisher*: Esta herramienta permite crear aplicaciones de WEB.

## Paso 10-Población

Este paso involucra nada más que la ejecución de los programas previamente desarrollados. Los puntos involucrados son los siguientes:

- Frecuencia de población:

Para el caso particular de este trabajo la frecuencia con la que se poblará el warehouse será mensual ya que los consumos realizados por los afiliados a las obras sociales interesa analizarlo mes a mes.

Esta afirmación no es al azar sino que resulta de la experiencia obtenida en el relevamiento realizado a las personas encargadas de analizar estos datos

- Depuración de los datos poblados

No se realizará una exhaustiva depuración de los datos poblados inmediatamente, ya que no es de interés analizar el volumen de consumos tan precisamente, la

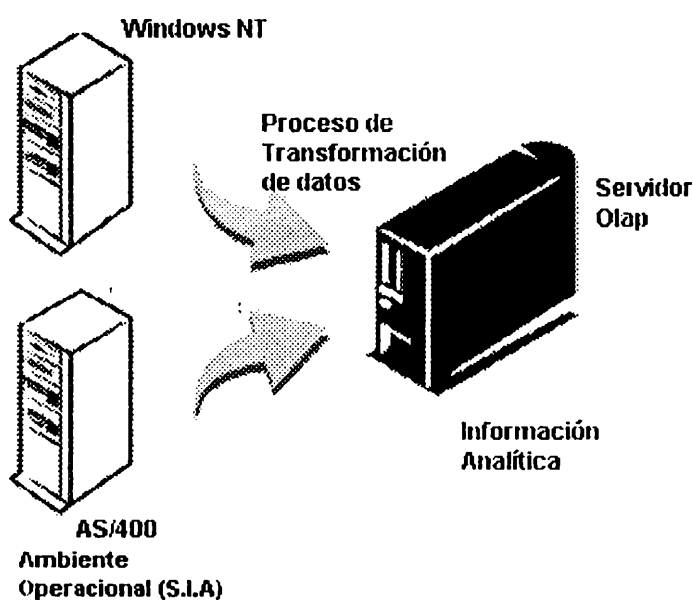
depuración de dichos datos quedará pendiente para el momento en que realicen análisis más precisos .

- Envejecimiento de los datos poblados :

A medida que pasa el tiempo se ejecutarán programas que realicen sumalizaciones de los datos ya que se van analizar datos por cuatrimestres, semestres, anuales, etc., es decir, esto quiere significar que a medida que los datos envejecen ya no es necesario tener un detalle exhaustivo de ellos como ser diario por ejemplo.

- Manejo de múltiples niveles de granularidad

Los niveles de granularidad manejados en este trabajo permiten realizar análisis de información, Mensual, trimestral, cuatrimestral, anual, o por épocas estacionarias.



---

## Conclusiones

---

Este trabajo pretende introducir a sus lectores el conocimiento básico y necesario sobre Data Warehousing. Para cumplir con este propósito hemos dividido el trabajo en varias secciones, cada una de ellas con un objetivo distinto: la primera ha intentando brindar una breve introducción del tema para poner al lector al tanto de la necesidad y características de la problemática que resuelve la tecnología de Data Warehousing, la segunda sección entrega los conceptos teóricos fundamentales que hay detrás de un data warehouse, por último la tercera sección ejemplifica la metodología de diseño de un Data Warehouse con un ejemplo concreto basándonos en una aplicación real.

De la experiencia obtenida en este trabajo podemos mencionar algunas ventajas y también desventajas que nos brinda la tecnología de Data Warehousing:

### **Ventajas:**

- **Simplicidad:** El Data Warehousing hace que los negocios sean simples ya que provee una imagen de la realidad comercial mediante la integridad de varios datos. Permite que los sistemas existentes continúen operando, consolida datos inconsistentes de varios sistemas en un conjunto coherente y obtiene beneficios de información vital de las operaciones actuales, éstas se pueden monitorear y comparar con operaciones pasadas, se pueden hacer predicciones de operaciones futuras y se puede disponer de nuevos procesos comerciales.
- **Datos de mejor calidad:** esto incluye la consistencia, exactitud y documentación de los datos con los cuales se puede realizar una mejor toma de decisión a través del análisis con herramientas OLAP y Data mining.
- **Rápido acceso:** los datos necesarios se encuentran ubicados en un lugar de manera que el tiempo de respuesta de los sistemas debería reducirse de forma significativa.

- **Facilidad de uso:** las consultas de los usuarios no interfieren con las operaciones normales porque un Data Warehouse permite un fácil acceso a los datos corporativos sin degradar la base de datos operacional tomando los datos operacionales y ubicándolos en una base de datos separada.
- **Operación para soporte de la toma de decisiones separada de la operación del ambiente de producción:** los Data Warehouses están contruidos para poder separar los datos transaccionales actualizados continuamente de los datos más estáticos e históricos requeridos para análisis comerciales. De este modo los gerentes y analistas pueden usar datos históricos sin degradar las tareas de producción.
- **Ventaja competitiva:** el Data warehousing maneja mejor y utiliza el conocimiento corporativo que a su vez ayuda al negocio a ser más competitivo, entender mejor a los clientes, y satisfacer más rápidamente las demandas de mercado. Por lo tanto este beneficio puede justificar el alto costo.
- **Manejo del flujo de información:** el Data Warehousing maneja una gran cantidad de datos provenientes de varias fuentes de datos operacionales y administra el flujo de la información más que simplemente recolectar los datos.
- **Permite procesamiento paralelo:** los usuarios pueden realizar consultas que antes resultaban de procesamiento muy intensivo para poder responder, y el Data Warehouse puede manejar más clientes, usuarios, transacciones, consultas y mensajes.
- **Soporta demandas de mayor performance en ambientes cliente/servidor,** provee escalabilidad ilimitada y, por lo tanto, mayor relación de precio/performance.
- **Independencia de la plataforma:** el Data Warehouse puede construirse sobre cualquier plataforma, desde una PC a un mainframe, aunque muchos están eligiendo servidores UNIX y corriendo su data warehousing en un ambiente cliente/servidor.
- **Infraestructura de computación:** ayuda a la organización a crear una infraestructura de computación que pueda soportar cambios en los sistemas de computación y en las estructuras de negocios.
- **Valor cuantitativo:** el Data Warehouse provee las métricas cuantitativas necesarias para establecer líneas de procesas comerciales que se derivan de datos históricos y permiten a los gerentes medir el progreso.

- Seguridad: los usuarios del Data Warehouse no pueden consultar directamente a las bases de datos de producción, de esta manera se mejora la seguridad de éstas como así también su productividad.

### **Desventajas**

- Complejidad y anticipación en el desarrollo: la construcción de un Data Warehouse requiere de un sentido de anticipación acerca de las futuras formas de usar los registros recolectados. Los desarrolladores necesitan estar al tanto de las necesidades cambiantes constantemente del negocio de su compañía y de las capacidades del hardware y software emergente. La evolución que debe sufrir el warehouse para satisfacer las necesidades crecientes de los usuarios, tanto en volumen como en complejidad, hace a su desarrollo aún más complejo. Hay también dificultades en la elección de los productos correctos. En resumen, desarrollar una base de datos tan grande requiere de un experto.
- Tiempo de construcción: si no se posee un fuerte apoyo ejecutivo los directores de los sistemas de información otros que deseen desarrollar un warehouse pueden gastar una gran cantidad de tiempo tratando de justificar la necesidad. El tiempo estimado para la construcción de un Data Warehouse oscila entre los 2 y 3 años.
- Costo de la construcción: una razón por la cual los data warehouses son tan caros es que los datos deben ser movidos migrados o copiados desde las bases de datos existentes, a veces manualmente, y éstos se deben traducir a un formato común.
- Falta de API (application programming interface): aún faltan API's u otros estándares para soportar los procesos del warehouse, como por ejemplo, las interfaces de ODBC.
- Capacitación del usuario final: es necesario concientizar a los empleados, que deben estar preparados para capitalizar el análisis innovador de datos que proveen los data warehouses; dichos usuarios requieren un entrenamiento intensivo
- Dificultad en el ambiente de base de datos distribuidas: porque el Data Warehouse es un método de unir datos dispares, su naturaleza es centralizada en sí misma.

## Trabajo a futuro

Debido al alcance que tuvo esta tesis, creemos conveniente mencionar dos posibles extensiones a este Trabajo de Grado. A continuación pasaremos a detallar cada una de estas posibles realizaciones justificando la conveniencia de cada una de ellas.

El Data Warehousing es un nuevo campo dentro de las ciencias de la computación, es difícil estimar qué nuevos desarrollos probablemente lo afectará. Existe una realidad que parece que se extenderá en el tiempo y permanecerá vigente con respecto a esta tecnología y es que la calidad de la información que ofrecen los Data Warehouses para soportar a la toma de decisiones ejecutivas o gerenciales es muy superior a la brindada por las herramientas de consultas y reportes tradicionales.

Otro de los desarrollos más importantes entre las ciencias de la computación desde el advenimiento de las computadoras personales es la explosión de Internet y las aplicaciones basadas en la Web. La comunidad de los negocios se ha volcado en forma masiva al tren de Internet. Uno de los campos más excitantes en la industria de la computación hoy es el desarrollo de aplicaciones Intranet. Las intranets son redes comerciales privadas que están basadas en estándares de Internet, aunque están diseñadas para ser usadas en forma interna. La tendencia Internet/Intranet tiene implicaciones muy importantes para las aplicaciones de DataWarehousing. Primero, los Data Warehouses pueden estar disponibles para todo el mundo en redes publica/privadas a un costo mucho más bajo. Esta disponibilidad minimiza la necesidad de replicar datos en distintas ubicaciones geográficas. De esta manera la comunidad empresarial y de negocios encargada de realizar la toma de decisiones puede realizar su tarea de consulta y análisis de la información brindada por sus sistemas de Data Warehousing sin importar el lugar físico en el que se encuentre, desde su casa en su P.C., desde su oficina en Buenos Aires, desde un avión en su laptop o desde su oficina ubicada en New York.

Segundo, este estándar ha permitido que el servidor de web provea una capa media donde poder realizar el análisis antes de ser usado por el usuario de web.

El poder del hardware y del software junto con la disponibilidad de herramientas de análisis y de reporting de fácil uso a precios accesibles han jugado el rol más importante en la evolución de los Data Warehouses.

Este trabajo presentó el diseño de un Data Warehouse específico, queda entonces planteada la base para la futura realización de un próximo trabajo cuyo alcance sea la implementación concreta de tal diseño complementando tal implementación con la posibilidad de tener acceso a la información provista por este Data Warehouse desde las mencionadas redes privadas Intranets como también desde Internet,

teniendo en cuenta que para tal desarrollo obviamente se debe adquirir el conocimiento necesario que involucre el tema.

La segunda opción que creemos que puede ser considerada como desarrollo para un nuevo Trabajo de Grado a partir del presente es el tema de Data Mining ya que un ambiente de Data Warehouse es ideal para la aplicación de técnicas de Data Mining.

El Data Mining encuentra respuestas a preguntas de negocios que nadie habría pensado en realizar. Descubre información dentro de los Data Warehouses que las consultas y reportes no pueden revelar efectivamente. Los beneficios potenciales del Data Mining son enormes si se eligen las herramientas adecuadas y se las usa efectivamente. Estas aplicaciones se pueden convertir en el fundamento de las estrategias de negocio de la organización- determinando riesgo crediticio, detección de fraudes, manejo de garantías de producto y definición de nuevos productos y servicios de telecomunicaciones.

Las consultas de bases de datos tradicionales están diseñadas para dar respuesta a preguntas simples como ¿Cuáles fueron mis ventas en Enero de 1995 en la región noroeste?, el análisis multidimensional (OLAP) permite a los usuarios hacer consultas mucho más complejas de modo tal de poder comparar las ventas relativas a planes por cuatrimestres y regiones para los dos años anteriores. Pero en ambos casos, los resultados son meramente valores extraídos o una agregación de valores.

El Data Mining profundiza mucho más en las bases de datos. Las herramientas de Data Mining encuentran patrones en los datos e infieren reglas sobre ellos. Estos patrones y reglas pueden ser usados para guiar en la toma de decisiones y pronosticar el efectos de tales decisiones. El Data Mining puede acelerar el análisis enfocando la atención en las variables más importantes del negocio.

Estos patrones se podrían encontrar con una serie de consultas contra los datos, pero el Data Mining permite a los usuarios explorar un rango mucho más amplio de posibilidades que el más sofisticado conjunto de consultas.

El Data Mining está creciendo ya que las organizaciones están recolectando cada vez más datos de sus negocios. La enorme caída en los precios de almacenamiento ha hecho factible mantener enormes cantidades de datos en línea. Algunos de estos datos provienen de los sistemas de procesamiento de transacciones en línea(OLTP), pero gran parte de ellos son el resultado de los sistemas instalados en años recientes para capturar todos los detalles de una transacción que ayude a las compañías a entender mejor qué quieren y hacen realmente, no qué dicen.

Por ejemplo, algunas cadenas de mercados están promocionando a sus clientes tarjetas que les den descuentos cuando se presente la misma en la línea de cajas. El



mercado puede decir a través de su sistema de escáner no solamente qué hay en el “changuito” sino también quién lo compró.

La caída dramática en la relación costo/performance de los sistemas de computación ha permitido que muchas organizaciones comiencen a aplicar los algoritmos complejos que se usan en las técnicas de Data Mining.

El surgimiento del Data Warehousing también ha reducido las barreras existentes para el Data Mining. En el pasado, a menudo era necesario recolectar los datos, depurarlos y mezclarlos. Ahora en muchos casos, la información se encuentra almacenada en Data Warehouses pidiendo a gritos “Úsenme, Úsenme”.

Hay cinco tipos comunes de información que se pueden producir con herramientas de Data Mining: asociaciones, secuencias, clasificaciones, clusters y pronósticos.

Las asociaciones surgen cuando las ocurrencias se vinculan en un evento. Por ejemplo, un estudio de mercado revela que cuando se compra una determinada cantidad de unidades de un determinado producto en un determinado período de tiempo entonces se sabe que la compra de este producto puede involucrar la compra de otro complementario de este, en consecuencia los gerentes pueden evaluar la ganancia de una promoción para estos productos en esa época del año.

En las secuencias los eventos son vinculados en el tiempo. Si se compra una casa, entonces el 45% de las veces se comprará un horno dentro de un (1) mes y el 60 % de las veces se comprará un nuevo refrigerador dentro de las dos (2) semanas.

La clasificación es probablemente la actividad más común de Data Mining hoy en día. Reconoce patrones que describen el grupo al que pertenece un ítem. Esta tarea la realiza examinando ítems existentes que ya han sido clasificados e infiriendo un conjunto de reglas. La clasificación puede ayudar a descubrir las características de los clientes que sean propensos a abandonar el negocio, es decir aquellas que por características de algún tipo se pueda inferir que son potenciales emigrantes como clientes de la empresa.

El clustering está relacionado con la clasificación, pero difiere en que no hay grupos que ya se hayan definido. Con el uso de Clustering, la herramienta de Data Mining descubre diferentes agrupamientos dentro de los datos. Esto puede aplicarse a problemas variados como la detección de defectos de fabricación o encontrar grupos que tengan afinidad a cierta tarjeta bancaria.

Todas estas aplicaciones pueden involucrar predicciones, tal como saber si un cliente va a renovar su suscripción. El quinto tipo de aplicación, pronóstico o



forecasting, es una forma diferente de predicción. Estima el valor futuro de ciertas ventas de tipo variables continuas basándose en patrones dentro de los datos.

Por último existen cuatro tipos de herramientas usadas en Data Mining: redes neuronales, arboles de decisión, inducción de reglas y visualización de datos. Algunas herramientas están basadas en combinaciones de estos métodos.

Todo lo expuesto previamente junto con la base que deja este trabajo de Grado en el diseño del Data Warehouse da lugar a un nuevo desarrollo para otro Trabajo que incluya el estudio de las distintas técnicas de Data Mining para obtener información de la implementación del diseño del Data Warehouse que planteó esta tesis.

---

## Anexo

---

Las siguientes tablas muestran una posible definición de los atributos de nuestras tablas de dimensión y fact tables (tablas de hechos). Para cada una de ellas se indica:

- *Tipo de Tabla*: Dimensión ó Fact Table
- *Nombre del campo o atributo*
- *Tipo* : Carácter, Numérico, Fecha
- *Origen del dato*: Tabla y atributo del modelo operacional
- *Derivado / Primitivo* : Clasificación de los atributos
- *Descripción* : Un breve comentario del atributo.

**Tipo: Dimensión**  
**Nombre: Perfil de Afiliado**

Nro	Campo	Tipo de Dato	Origen del Dato	Der / Prim	Descripción
1	Clave_Afiliado	Númerico(10)	-	Derivado	
2	Estado	Caracter	-	Derivado	Embarazo, Post_parto, Discapacitado
3	Categorización	Caracter	-	Derivado	Titular, Dependiente del titular
4	Edad	Númerico	-	Derivado	Edad del perfil
5	Ingreso_promedio	Númerico	-	Derivado	Monto que representa los ingresos del perfil
6	Sexo	Caracter	-	Derivado	Sexo
7	Estado_civil	Caracter	-	Derivado	Casado, Soltero, Viudo,
8	Cant_hijos	Númerico	-	Derivado	Cantidad de hijos a cargo
9	Nivel_estudio	Caracter	-	Derivado	Primario, Secundario, Terciario, Universitario
10	Relación_laboral	Caracter	-	Derivado	Relación de dependencia, Independiente
11	Ocupación	Caracter	-	Derivado	Empleado, Profesional, Jubilado
12	Cargo_desempeñado	Caracter	-	Derivado	Empleado, Jefe, Subjefe, Gerente, Dueño

**Tablas Involucradas : BACatTi, BACIMi2, BACIMie, BAGfMie, BAGFVig**

**Tipo: Dimensión**  
**Nombre: Regiones Geográficas**

Nro	Campo	Tipo de Dato	Origen del Dato	Der / Prim	Descripción
1	Clave_Geografía	Númerico(10)	-		
2	Provincia	Caracter(1)	Pcias.PciaCodigo	Primitivo	Código de la Provincia
3	Descripción_provincia	Caracter(20)	Pcias.PciaDescri	Primitivo	Descripción de la Provincia
4	Localidad	Númerico(5)	Localida.LocaCodigo	Primitivo	Código de la Localidad
5	Descripción_localidad	Caracter(30)	Localida.LocaDescri	Primitivo	Descripción de la Localidad
6	Zona_geográfica	Númerico(5)	-	Derivado	Código de la region Geográfica
7	Descripción_zona	Caracter(30)	-	Derivado	Descripción de la Region Geográfica
8	Barrio	Númerico(5)	BarLocal.BarrCodigo	Primitivo	Código de Barrio
9	Descripción_barrio	Caracter(30)	BarLocal.BarrDescri	Primitivo	Descripción del Barrio
10	Delegación	Númerico(5)	Delegac.DeleNumDel	Primitivo	Código de la Delegación
11	Descripción_delegación	Caracter(30)	Delegac.DeleNomDel	Primitivo	Descripción de la Delegación

**Tipo: Dimensión**

**Nombre: Responsable de Facturación**

Nro	Campo	Tipo de Dato	Origen del Dato	Der / Prim	Descripción
1	Clave_Responsable	Númerico(10)	-	-	
2	Responsable	Númerico(6)	Proveedo.ProvNumInt	Primitivo	Código del Responsable o Proveedor
3	Nombre del responsable	Carácter(35)	Proveedo.ProvRazSoc	Primitivo	Nombre del Responsable o Proveedor
4	Categorización	Númerico(3)	Proveedo.ProcCodigo	Primitivo	Código de Categorización
5	Descripción categorización	Carácter(40)	ProCalif.ProcDescri	Primitivo	Hospital, Agronomía, Particular
6	Especialización	Carácter(3)	PrRAteenc.EspeCodigo	Primitivo	Código de Especialización
7	Descripción especialización	Carácter(35)	Especial.EspeDescri	Primitivo	Descripción del la especialización
8	Tipo convenio	Carácter(2)	CatConve.CatCodCat	Primitivo	Tipo de Convenio del Prestador
9	Vigencia convenio	Fecha(8)	CatConve.CoviFecVig	Primitivo	Vigencia del convenio
10	Fecha de Alta	Fecha(8)	-	-	Fecha de alta del registro

**Tipo: Dimensión**

**Nombre: Prótesis y Optica**

Nro	Campo	Tipo de Dato	Origen del Dato	Der / Prim	Descripción
1	Clave_ProOrOp	Númerico(10)	-	-	
2	Codigo elemento	Númerico(7)	Elemento.ElemNumInt	Primitivo	Código del Tipo del Elemento de Prótesis u Optica
3	Descripcion cod elemento	Carácter(30)	Elemento.ElemDescri	Primitivo	Descripción del Elemento de Prótesis u Optica
4	Código grupo	Númerico(7)	Elemento.ElgrNumInt	Primitivo	Código del Grupo del elemento de Prótesis u Optica
5	Descripción cod grupo	Carácter(30)	Elemento.ElgrDescri	Primitivo	Descripción del Grupo del Elemento de Prótesis u Optica
6	Código tamaño	Númerico(3)	Elemento.EITaCodigo	Primitivo	Código del Tamaño del Elemento de Prótesis u Optica
7	Descripción cod tamaño	Carácter(30)	EITamano.EITaDescri	Primitivo	Descripción del Tamaño del Elemento de Prótesis u Optica
8	Código material	Númerico(5)	Elemento.EIMtCodigo	Primitivo	Código del Material delElemento de Prótesis u Optica
9	Descripción cod material	Carácter(30)	EIMaterial.EIMtDescri	Primitivo	Descripción del Material del Elemento de Prótesis u Optica
10	Descripción técnica elemento	Carácter(45)	Elemento.ElemDesTec	Primitivo	Descripción del Técnica del Elemento de Prótesis u Optica
11	Código area	Númerico(7)	-	-	Código de Area del Elemento de Prótesis u Optica
12	Descripción cod area	Carácter(30)	-	-	Descripción del Area del Elemento de Prótesis u Optica
13	Código subgrupo	Númerico(7)	Elemento.EIMtCodigo	Primitivo	Código de Subgrupo del Elemento de Prótesis u Optica
14	Descripción cod subgrupo	Carácter(30)	EISubgru.ElsuDescri	Primitivo	Descripción de Subgrupo del Elemento de Prótesis u Optica
15	Código tipo	Númerico(7)	Elemento.EltiCodigo	Primitivo	Código del Elemeto de Prótesis u Optica
16	Descripción cod tipo	Carácter(30)	EITipos.EltiDescri	Primitivo	Descripción del Tipo del Elemento de Prótesis u Optica
17	Sexo asociado	Carácter(1)	-	Derivado	Código de Sexo

Tipo: Dimensión

Nombre: Tiempo

Nro	Campo	Tipo de Dato	Origen del Dato	Der / Prim	Descripción
1	Clave_tiempo	Númerico(10)	-	Primitivo	Clave de acceso a la table
2	AñoMes	Caracter(6)	-	Primitivo	Año y mes en que se realizó un consumo
3	Año	Caracter(4)	-	Primitivo	Mes en que se realizó un consumo
4	EpocaEstacional	Caracter(10)	-	Primitivo	Epoca estacional en que se realizó un consumo
5	VacacionesEscolares	Caracter(1)	-	Primitivo	S/N si el consumo se realizó en vacaciones

Tipo: Dimensión

Nombre: Plan\_Cobertura

Nro	Campo	Tipo de Dato	Origen del Dato	Der / Prim	Descripción
1	Clave_plan	Númerico(10)	-	Primitivo	Clave de acceso a la table
2	Codigo_plan	Númerico(6)	PlCabece.PlCaNumPla	Primitivo	Código de Plan
3	Descr_plan	Caracter(25)	PlCabece.PlCaDesAmp	Primitivo	Descripción del plan
4	FechaVigPlan	Fecha(8)	PlCabece.PlCaFecVig	Primitivo	Fecha de vigencia del plan
5	FechaInhabPlan	Fecha(8)	PlCabece.PlCaFecBaj	Primitivo	Fecha de inhabilitación del plan
6	ReqAutoriz	Caracter(1)	PlCabece.PlCaReqAut	Primitivo	Requiere Autorización
7	PerReintegro	Caracter(1)	PlCabece.PlCaPerRein	Primitivo	Permite Reintegro
8	TopeMontoRein	Númerico(10.2)	Pitem.PltemMonRei	Primitivo	Tope Monto Reintegro
9	TopeMontoAut	Númerico(10.2)	Pitem.PltemMonAut	Primitivo	Tope Monto Autorización
10	Programa	Númerico(2)	HACabece.HACaCodGra	Primitivo	Código de Programa
11	Servicio	Númerico(3)	PaServic.PaSeCodSer	Primitivo	Código de Servicio
12	Descr_Servicio	Caracter(30)	PaServic.PaSeDesSer	Primitivo	Descripción del servicio
13	Gravamen	Númerico(2)	Gravamen.GravCodGra	Primitivo	Código de gravámen
14	Descr_Gravamen	Caracter(30)	Gravamen.GravDesGra	Primitivo	Descripción del gravamen
15	Tipo_Periodo_Reintegro	Caracter(1)	Pitem.PltePeriRe	Primitivo	Tipo Periodo Reintegro
16	Tipo_Periodo_Autorizacion	Caracter(1)	Pitem.PltePerAut	Primitivo	Tipo Periodo Autorización
17	Cantidad_Periodo_Reintegro	Númerico(3)	Pitem.PlteCapRei	Primitivo	Cantidad Periodo Reintegro
18	Base_Valorización	Caracter(1)	Pitem.BaseValori	Primitivo	Base Valorización
19	Valor_Reintegro	Númerico(6.2)	Pitem.PlteValRei	Primitivo	Valor Reintegro
20	Tipo_De_Operación	Caracter(1)	Pitem.PlteTipOpe	Primitivo	Tipo De Operación
21	Cantidad_Periodo_Autorización	Númerico(3)	Pitem.PlteCapAut	Primitivo	Cantidad Periodo Autorización
22	Tope_Cantidad_Autorizaciones	Númerico(3)	Pitem.PlteAutCan	Primitivo	Tope Cantidad Autorizaciones
23	Tope_Cantidad_Reintegros	Númerico(3)	Pitem.PlteReiCan	Primitivo	Tope Cantidad Reintegros
24	Requiere_Expediente	Caracter(1)	Pitem.PlteReqExp	Primitivo	Requiere Expediente
25	Requiere_Presupuesto	Caracter(1)	Pitem.PlteReqPre	Primitivo	Requiere Presupuesto
26	Nivel_Autorización	Caracter(3)	Pitem.PlteNivAut	Primitivo	Nivel Autorización

**Nombre: Prácticas Médicas**

Nº	Campo	Tipo de Dato	Origen del Dato	Der / Prim	Descripción
1	Clave_Práctica	Númerico(10)	-		
2	Tipo_práctica	Caracter(30)	-	Derivado	(ambulatoria, internación)
3	Sección_nomenclador	Númerico(2)	Nomenclca.SecnCodigo	Primitivo	Código de Sección del Nomenclador
4	Desc_sección_nomenclador	Caracter(30)	SecNomen. SecnDescri	Primitivo	Descripción del código de Sección del Nomenclador
5	Capitulo_nomenclador	Númerico(2)	Nomenclca.SecaNumCap	Primitivo	Código de Capítulo del Nomenclador
6	Desc_Capitulo_nomenclador	Caracter(30)	SPCapit . SPCapDescr	Primitivo	Descripción del código de Capítulo del Nomenclador
7	Subcapitulo_nomenclador	Númerico(2)	Nomenclca.NomeNumSub	Primitivo	Código de Subcapítulo del Nomenclador
8	Desc_Subcapitulo_nomenclador	Caracter(30)	-	Primitivo	Descripción del código de Subcapítulo del Nomenclador
9	Secuencia	Númerico(2)	Nomenclca.NomeNumSec	Primitivo	Secuencia
10	Subitem_nomenclador	Númerico(2)	Nomenclca.SubiCodSub	Primitivo	Código de Subitem del Nomenclador
11	Desc_subitem_nomenclador	Caracter(30)	SubItems.SubiDesSub	Primitivo	Descripción del código de Subitem del Nomenclador
12	Origen_nomenclador	Númerico(2)	Nomenclca.OrinCodigo	Primitivo	Código de Origen del Nomenclador
13	Desc_origen_nomenclador	Caracter(15)	OrinNomen.OrinDesOri	Primitivo	Descripción del código de Origen del Nomenclador
14	Tipo_nomenclador	Númerico(2)	NomTipos.TipnCodigo	Primitivo	Código de Tipo del Nomenclador
15	Desc_tipo_nomenclador	Caracter(60)	DesNomen.DesDesNom	Primitivo	Descripción del código de Tipo del Nomenclador
16	Coseguro	Númerico(1)	Pitem.TiCsTipCos	Primitivo	Código de Coseguro
17	Tipo_coseguro	Caracter(30)	Pitem.TiCsDescri	Primitivo	Descripción del Coseguro
18	Carencia	Caracter(1)	Pitem. PiteCarenc	Primitivo	Código de Carencia
19	Desc_carencia	Caracter(30)	-		Descripción de la Carencia
20	Codigo_plan	Númerico(6)	PICabece.PICaNumPla	Primitivo	Código de Plan
21	Descr_plan	Caracter(25)	PICabece.PICaDesAmp	Primitivo	Descripción del plan
22	FechaVigPlan	Fecha(8)	PICabece.PICaFecVig	Primitivo	Fecha de vigencia del plan
23	FechaInhabPlan	Fecha(8)	PICabece.PICaFecBaj	Primitivo	Fecha de inhabilitación del plan
24	ReqAutoriz	Caracter(1)	PICabece.PICaReqAut	Primitivo	Requiere Autorización
25	PerReintegro	Caracter(1)	PICabece.PICaPerRein	Primitivo	Permite Reintegro
26	TopeMontoRein	Númerico(10.2)	Pitem.PitemMonRei	Primitivo	Tope Monto Reintegro
27	TopeMontoAut	Númerico(10.2)	Pitem.PitemMonAut	Primitivo	Tope Monto Autorización
28	Programa	Númerico(2)	HACabece.HACaCodGra	Primitivo	Código de Programa
29	Servicio	Númerico(3)	PaServic.PaSeCodSer	Primitivo	Código de Servicio
30	Descr_Servicio	Caracter(30)	PaServic.PaSeDesSer	Primitivo	Descripción del servicio
31	Gravamen	Númerico(2)	Gravamen.GravCodGra	Primitivo	Código de gravámen
32	Descr_Gravamen	Caracter(30)	Gravamen.GravDesGra	Primitivo	Descripción del gravamen
33	Tipo_Periodo_Reintegro	Caracter(1)	Pitem.PitePeriRe	Primitivo	Tipo Periodo Reintegro
34	Tipo_Periodo_Autorizacion	Caracter(1)	Pitem.PitePerAut	Primitivo	Tipo Periodo Autorizacion
35	Cantidad_Periodo_Reintegro	Númerico(3)	Pitem.PiteCapRei	Primitivo	Cantidad Periodo Reintegro
36	Base_Vvalorización	Caracter(1)	Pitem.BaseValori	Primitivo	Base Valorizacion
37	Valor_Reintegro	Númerico(6.2)	Pitem.PiteValRei	Primitivo	Valor Reintegro
38	Tipo_De_Operación	Caracter(1)	Pitem.PiteTipOpe	Primitivo	Tipo De Operacion
39	Cantidad_Periodo_Autorización	Númerico(3)	Pitem.PiteCapAut	Primitivo	Cantidad Periodo Autorizacion

Nombre: Prácticas Médicas

Nro	Campo	Tipo de Dato	Origen del Dato	Der / Prim	Descripción
40	Tope Cantidad Autorizaciones	Número(3)	Pitem. PiteAutCan	Primitivo	Tope Cantidad Autorizaciones
41	Tope Cantidad Reintegros	Número(3)	Pitem. PiteReCan	Primitivo	Tope Cantidad Reintegros
42	Requiere Expediente	Caracter(1)	Pitem. PiteReqExp	Primitivo	Requiere Expediente
43	Requiere Presupuesto	Caracter(1)	Pitem. PiteReqPre	Primitivo	Requiere Presupuesto
44	Nivel Autorización	Caracter(3)	Pitem. PiteNivAut	Primitivo	Nivel Autorización
45	Nivel Reintegros	Caracter(3)	Pitem. PiteNivRei	Primitivo	Nivel Reintegros
46	Fecha de Alta	Fecha(8)	-	-	Fecha de Alta

Tipo: Dimensión

Nombre: Medicamentos

Nro	Campo	Tipo de Dato	Origen del Dato	Der / Prim	Descripción
1	Clave medicamento	Número(10)	-	Primitivo	Clave de acceso a la tabla
2	MediCodMed	Número(7)	Medicame.MediCodMed	Primitivo	Código de Medicamento
3	MediNomMed	Caracter(30)	Medicame.MediNomMed	Primitivo	Nombre fantasía del medicamento
4	MediConcan	Número(10.2)	Medicame.MediConcan	Primitivo	Concentración
5	MediUnioCon	Caracter(4)	Medicame.MediUnioCon	Primitivo	Unidad de concentración
6	MediCodFor	Caracter(4)	Medicame.FormCodFor	Primitivo	Código de Forma
7	MediCodZona	Caracter(4)	Medicame.ZonaCodZon	Primitivo	Zona Aplicación / Vía Administración
8	MediCodPre	Caracter(4)	Medicame.PremCodPre	Primitivo	Presentación
9	MediUniPre	Caracter(4)	Medicame.MediUniPre	Primitivo	Unidad de presentación
10	MediCanMed	Número(10.2)	Medicame.MediCanMed	Primitivo	Cantidad de presentación
11	MediVolPre	Número(9.3)	Medicame.MediVolPre	Primitivo	Contenido de Presentación
12	MediUniVol	Caracter(4)	Medicame.MediUniVol	Primitivo	Unidad de Volumen de Presentación
13	MediCodOrMed	Número(2)	Medicame.OrmeCodigo	Primitivo	Origen del Medicamento
14	MediCodLab	Caracter(4)	Medicame.MediCodLab	Primitivo	Código de laboratorio
15	MediCanDos	Número(3)	Medicame.MediCanDos	Primitivo	Cantidad de Aplicación
16	MediUniDos	Caracter(4)	Medicame.MediUniDos	Primitivo	Unidad de medida de dosis
17	MediNumTro	Número(8)	Medicame.MediNumTro	Primitivo	Número de troquel
18	MediDesPre	Caracter(40)	Medicame.MediDesPre	Primitivo	Presentación Original
19	MediCodUsu	Caracter(10)	Medicame.MediCodUsu	Primitivo	Código de usuario
20	MediCodSexo	Caracter(1)	Medicame.SexoCodigo	Primitivo	Código de sexo
21	MediEdaDes	Número(2)	Medicame.MediEdaDes	Primitivo	Edad Desde
22	MediEdaHas	Número(2)	Medicame.MediEdaHas	Primitivo	Edad Hasta
23	MediCanFra	Número(5.2)	Medicame.MediCanFra	Primitivo	Cantidad Mínima de fraccionamiento
24	MediCodRub	Caracter(1)	Medicame.MediCodRub	Primitivo	Código de rubro del medicamento
25	MediDesRub	Caracter(50)	OMS Rubro	Primitivo	Descripción de rubro
26	MediCodNac	Caracter(1)	Medicame.MediCodNac	Primitivo	Nacional o Importado
27	MediReqTroq	Caracter(1)	Medicame.MediReqTroq	Primitivo	Requiere troquel

Tipo: Dimensión

Nombre: Diagnósticos

Nro	Campo	Tipo de Dato	Origen del Dato	Der / Prim	Descripción
1	Clave_diagnostico	N Numérico(10)	-	Primitivo	Clave de acceso a la tabla
2	Codigo_rubro	Caracter(3)	OMSRubro.OmRuCodigo	Primitivo	Código dr rubro
3	Descr_rubro	Caracter(50)	OMSRubro.OmRuDescri	Primitivo	Descripción de rubro
4	CapitOMS	N Numérico(3)	OmCapit.OmCaCodigo	Primitivo	Capítulo O.M.S.
5	Descr_CapOMS	Caracter(50)	OmCapit.OmCaDescri	Primitivo	Descripción de capítulo O.M.S.
6	SubCapOMS	Caracter(4)	OmSubCap.OmSuCodigo	Primitivo	Subcapítulo de O.M.S.
7	Descr_SubCap	Caracter(60)	OmSubCap.OmSuDescri	Primitivo	Descripción de subcapítulo O.M.S.
46	Fecha de Alta	Fecha(8)	-	-	Fecha de Alta



**Tipo: Fuct Table**  
**Nombre: Consumos Central**

Nro	Campo	Tipo de Dato	Origen del Dato	Der / Prim
1	Clave tiempo	Número(10)	-	Derivado
2	Clave Geografía	Número(10)	-	Derivado
3	Clave Afiliado	Número(10)	-	Derivado
4	Clave Diagnóstico	Número(10)	-	Derivado
5	Clave Plan	Número(10)	-	Derivado
6	Clave Responsable	Número(10)	-	Derivado
7	Cantidad consumos	Número(10)	Campos: (14) + (15) + (16)	Derivado
8	Cantidad personas atendidas	Número(10)	EsCapitaEsCaCanInt + EsCapitaEsCaCanAmb	Derivado
9	Consumo promedio	Número(10)	Campos: (10) / (7)	Derivado
10	Total facturado	Número(20,2)	Campos: (17) + (18) + (19)	Derivado
11	Total liquidado	Número(20,2)	Campos: (20) + (21) + (22)	Derivado
12	Total bonificado	Número(20,2)	Campos: (23) + (24) + (25)	Derivado
13	Total debitado	Número(20,2)	Campos: (26) + (27) + (28)	Derivado
14	Cantidad prácticass	Número(10)	Esfactur.EsFaCanAmb+ Esfactur.EsFaCanInt	Primitivo
15	Cantidad medicamentos	Número(10)	CaReceta.CaReUnMed	Primitivo
16	Cantidad prótesis ortesis optica	Número(10)	ReDOptic.ReDOCanPre+ReDProte.ReDPCanPre	Primitivo
17	Total prácticass facturado	Número(12,2)	Esfactur.EsFaInfAmb+Esfactur.EsFaInfInt	Primitivo
18	Total medicamentos facturado	Número(12,2)	CaReceta.CaReMonTot	Primitivo
19	Total PrOrOp facturado	Número(12,2)	ReDOptic.ReDOMonPra + ReDProte.ReDPMonPra	Primitivo
20	Total prácticass liquidado	Número(12,2)	Esfactur.EsFaLiqAmb +Esfactur.EsFaLiqInt	Primitivo
21	Total medicamentos liquidado	Número(12,2)	CaReceta.CaReMonLiq	Primitivo
22	Total PrOrOp liquidado	Número(12,2)	(ReDOptic.ReDOMonPra - ReDOptic.ReDOMonRei) + (ReDProte.ReDPMonPra- ReDProte. ReDPMonRei)	Primitivo
23	Total prácticass bonificado	Número(12,2)	Esfactur.EsFaInfBon+Esfactur.EsFaInfBon	Primitivo
24	Total medicamentos bonificado	Número(12,2)	CaReceta.CaReMonDes	Primitivo
25	Total PrOrOp bonificado	Número(12,2)	ReDOptic.ReDOMonRei + ReDProte. ReDPMonRei	Primitivo
26	Total prácticass debitado	Número(12,2)	Campos: (17) - (20) - (23)	Derivado
27	Total medicamentos debitado	Número(12,2)	Campos: (18) - (21) - (24)	Derivado
28	Total PrOrOp debitado	Número(12,2)	Campos: (19) - (22) - (25)	Derivado
29	Promedio prácticass facturado	Número(12,2)	Campos: (17) / (14)	Derivado
30	Promedio medicamentos facturado	Número(12,2)	Campos: (18) / (15)	Derivado
31	Promedio PrOrOp facturado	Número(12,2)	Campos: (19) / (16)	Derivado
32	Promedio prácticass liquidado	Número(12,2)	Campos: (20) / (14)	Derivado

Tipo: Fact Table  
Nombre: Consumos Central

Nro	Campo	Tipo de Dato	Der / Prim	Descripción
1	Clave_tiempo	Número(10)	Derivado	
2	Clave_Geografía	Número(10)	Derivado	
3	Clave_Afiliado	Número(10)	Derivado	
4	Clave_Diagnóstico	Número(10)	Derivado	
5	Clave_Plan	Número(10)	Derivado	
6	Clave_Responsable	Número(10)	Derivado	
7	Cantidad_consumos	Número(10)	Derivado	La suma de los consumos de los afiliados
8	Cantidad_personas_atendidas	Número(10)	Derivado	La cantidad de personas atendidas
9	Consumo_promedio	Número(10)	Derivado	Total Facturado / cantidad de personas atendidas
10	Total_facturado	Número(20,2)	Derivado	La suma de los totales facturados de prácticas , medicamentos y prótesis/óptica
11	Total_liquidado	Número(20,2)	Derivado	La suma de los totales liquidado de prácticas , medicamentos y prótesis/óptica
12	Total_bonificado	Número(20,2)	Derivado	La suma de los totales bonificado de prácticas , medicamentos y prótesis/óptica
13	Total_debitado	Número(20,2)	Derivado	La suma de los totales debitado de prácticas , medicamentos y prótesis/óptica
14	Cantidad_prácticass	Número(10)	Primitivo	La cantidad de practicas realizadas por los afiliados
15	Cantidad_medicamentos	Número(10)	Primitivo	La cantidad de medicamentos recetados a los afiliados
16	Cantidad_prótesis_ortesis_optica	Número(10)	Primitivo	La cantidad de prótesis y óptica utilizados por los afiliados
17	Total_prácticass_facturado	Número(12,2)	Primitivo	Monto facturado correspondiente a las prácticas médicas
18	Total_medicamentos_facturado	Número(12,2)	Primitivo	Monto facturado correspondiente a los medicamentos
19	Total_PrOrOp_facturado	Número(12,2)	Primitivo	Monto facturado correspondiente a las prótesis y óptica
20	Total_prácticass_liquidado	Número(12,2)	Primitivo	Monto liquidado correspondiente a las prácticas médicas
21	Total_medicamentos_liquidado	Número(12,2)	Primitivo	Monto liquidado correspondiente a los medicamentos
22	Total_PrOrOp_liquidado	Número(12,2)	Primitivo	Monto liquidado correspondiente a las prótesis y óptica
23	Total_prácticass_bonificado	Número(12,2)	Primitivo	Monto bonificado correspondiente a las prácticas médicas
24	Total_medicamentos_bonificado	Número(12,2)	Primitivo	Monto bonificado correspondiente a los medicamentos
25	Total_PrOrOp_bonificado	Número(12,2)	Primitivo	Monto bonificado correspondiente a las prótesis y óptica
26	Total_prácticass_debitado	Número(12,2)	Derivado	Monto debitado correspondiente a las prácticas médicas
27	Total_medicamentos_debitado	Número(12,2)	Derivado	Monto debitado correspondiente a los medicamentos
28	Total_PrOrOp_debitado	Número(12,2)	Derivado	Monto debitado correspondiente a las prótesis y óptica
29	Promedio_prácticass_facturado	Número(12,2)	Derivado	Monto promedio facturado correspondiente a las prácticas médicas
30	Promedio_medicamentos_facturado	Número(12,2)	Derivado	Monto promedio facturado correspondiente a los medicamentos
31	Promedio_PrOrOp_facturado	Número(12,2)	Derivado	Monto promedio facturado correspondiente a las prótesis y óptica
32	Promedio_prácticass_liquidado	Número(12,2)	Derivado	Monto promedio liquidado correspondiente a las prácticas médicas

Tipo: Fact Table

Nombre: Consumos Central

Nro	Campo	Tipo de Dato	Origen del Dato	Der / Prim
33	Promedio medicamentos liquidado	Número(12,2)	Campos: (21) / (15)	Derivado
34	Promedio PrOrOp liquidado	Número(12,2)	Campos: (22) / (16)	Derivado
35	Promedio prácticas bonificado	Número(12,2)	Campos: (23) / (14)	Derivado
36	Promedio medicamentos bonificado	Número(12,2)	Campos: (24) / (15)	Derivado
37	Promedio PrOrOp bonificado	Número(12,2)	Campos: (25) / (16)	Derivado
38	Promedio prácticas debitado	Número(12,2)	Campos: (26) / (14)	Derivado
39	Promedio medicamentos debitado	Número(12,2)	Campos: (27) / (15)	Derivado
40	Promedio PrOrOp debitado	Número(12,2)	Campos: (28) / (16)	Derivado

Tipo: Fact Table

Nombre: Consumos / Medicamentos

Nro	Campo	Tipo de Dato	Origen del Dato	Der / Prim
1	Clave tiempo	Número(10)	-	-
2	Clave Geografía	Número(10)	-	-
3	Clave Afiliado	Número(10)	-	-
4	Clave Diagnóstico	Número(10)	-	-
5	Clave Plan	Número(10)	-	-
6	Clave Responsable	Número(10)	-	-
7	Clave Medicamento	Número(10)	-	-
8	Cantidad consumos	Número(10)	CaReceta. CaReUN.UMed	Primitivo
9	Total facturado	Número(12,2)	CaReceta. CaReMonTot	Primitivo
10	Total liquidado	Número(12,2)	CaReceta. CaReMonLiq	Primitivo
11	Total bonificado	Número(12,2)	CaReceta. CaReMonDes	Primitivo
12	Total debitado	Número(12,2)	Campos: (9) - (10) - (11)	Derivado
13	Promedio facturado	Número(12,2)	Campos: (9) / (8)	Derivado
14	Promedio liquidado	Número(12,2)	Campos: (10) / (8)	Derivado
15	Promedio bonificado	Número(12,2)	Campos: (11) / (8)	Derivado
16	Promedio debitado	Número(12,2)	Campos: (12) / (8)	Derivado

Tipo: Fact Table

Nombre: Consumos Central

Nro	Campo	Tipo de Dato	Der / Prim	Descripción
33	Promedio_medicamentos_liquidado	Numérico(12,2)	Derivado	Monto promedio liquidado correspondiente a los medicamentos
34	Promedio_PrOrOp_liquidado	Numérico(12,2)	Derivado	Monto promedio liquidado correspondiente a las prótesis y óptica
35	Promedio_prácticas_bonificado	Numérico(12,2)	Derivado	Monto promedio bonificado correspondiente a las prácticas médicas
36	Promedio_medicamentos_bonificado	Numérico(12,2)	Derivado	Monto promedio bonificado correspondiente a los medicamentos
37	Promedio_PrOrOp_bonificado	Numérico(12,2)	Derivado	Monto promedio bonificado correspondiente a las prótesis y óptica
38	Promedio_prácticas_debitado	Numérico(12,2)	Derivado	Monto promedio debitado correspondiente a las prácticas médicas
39	Promedio_medicamentos_debitado	Numérico(12,2)	Derivado	Monto promedio debitado correspondiente a los medicamentos
40	Promedio_PrOrOp_debitado	Numérico(12,2)	Derivado	Monto promedio debitado correspondiente a las prótesis y óptica

Tipo: Fact Table

Nombre: Consumos / Medicamentos

Nro	Campo	Tipo de Dato	Der / Prim	Descripción
1	Clave_tiempo	Numérico(10)	-	-
2	Clave_Geografia	Numérico(10)	-	-
3	Clave_Afiliado	Numérico(10)	-	-
4	Clave_Diagnóstico	Numérico(10)	-	-
5	Clave_Plan	Numérico(10)	-	-
6	Clave_Responsable	Numérico(10)	-	-
7	Clave_Medicamento	Numérico(10)	-	-
8	Cantidad_consumos	Numérico(10)	Primitivo	La cantidad de medicamentos recetados a los afiliados
9	Total_facturado	Numérico(12,2)	Primitivo	Monto facturado correspondiente a los medicamentos
10	Total_liquidado	Numérico(12,2)	Primitivo	Monto liquidado correspondiente a los medicamentos
11	Total_bonificado	Numérico(12,2)	Primitivo	Monto bonificado correspondiente a los medicamentos
12	Total_debitado	Numérico(12,2)	Derivado	Monto debitado correspondiente a los medicamentos
13	Promedio_facturado	Numérico(12,2)	Derivado	Monto promedio facturado correspondiente a los medicamentos
14	Promedio_liquidado	Numérico(12,2)	Derivado	Monto promedio liquidado correspondiente a los medicamentos
15	Promedio_bonificado	Numérico(12,2)	Derivado	Monto promedio bonificado correspondiente a los medicamentos
16	Promedio_debitado	Numérico(12,2)	Derivado	Monto promedio debitado correspondiente a los medicamentos

Nombre: Consumos / Prótesis Ortesis Óptica

Nro	Campo	Tipo de Dato	Origen del Dato	Der / Prim
1	Clave tiempo	Número(10)	-	
2	Clave Geografía	Número(10)	-	
3	Clave Afiliado	Número(10)	-	
4	Clave Diagnóstico	Número(10)	-	
5	Clave Plan	Número(10)	-	
6	Clave Responsable	Número(10)	-	
7	Clave Prótesis_ortesis_optica	Número(10)	-	
8	Cantidad_prótesis_ortesis_optica	Número(10)	ReDOptic.ReDOCanPre+ReDProte.ReDPCanPre	Primitivo
9	Total_facturado	Número(12,2)	ReDOptic.ReDOMonPra + ReDProte.ReDPMonPra	Primitivo
10	Total_liquidado	Número(12,2)	(ReDOptic.ReDOMonPra ~ ReDOptic.ReDOMonRei) + (ReDProte.ReDPMonPra-ReDProte. ReDPMonRei)	Primitivo
11	Total_bonificado	Número(12,2)	ReDOptic.ReDOMonRei + ReDProte. ReDPMonRei	Primitivo
12	Total_debitado	Número(12,2)	Campos: (9) - (10) - (11)	Derivado
13	Promedio_facturado	Número(12,2)	Campos: (9) / (8)	Derivado
14	Promedio_liquidado	Número(12,2)	Campos: (10) / (8)	Derivado
15	Promedio_bonificado	Número(12,2)	Campos: (11) / (8)	Derivado
16	Promedio_debitado	Número(12,2)	Campos: (12) / (8)	Derivado

Tipo: Fact Table

Nombre: Consumos / Prácticas Médicas

Nro	Campo	Tipo de Dato	Origen del Dato	Der / Prim
1	Clave tiempo	Número(10)	-	
2	Clave Geografía	Número(10)	-	
3	Clave Afiliado	Número(10)	-	
4	Clave Diagnóstico	Número(10)	-	
5	Clave Plan	Número(10)	-	
6	Clave Responsable	Número(10)	-	
7	Clave Práctica	Número(10)	-	
8	Cantidad_consumos	Número(10)	Esfactor.EsFaCanAmb+ Esfactor.EsFaCanInt	Primitivo
9	Total_facturado	Número(12,2)	Esfactor.EsFalnAmb+Esfactor.EsFalnInt	Primitivo
10	Total_liquidado	Número(12,2)	Esfactor. EsFaliqAmb +Esfactor.EsFaliqInt	Primitivo
11	Total_bonificado	Número(12,2)	Esfactor.EsFalnBon+Esfactor.EsFalnBon	Primitivo
12	Total_debitado	Número(12,2)	Campos: (10) - (11) - (12)	Derivado
13	Promedio_facturado	Número(12,2)	Campos: (10) / (8)	Derivado
14	Promedio_liquidado	Número(12,2)	Campos: (11) / (8)	Derivado
15	Promedio_bonificado	Número(12,2)	Campos: (12) / (8)	Derivado
16	Promedio_debitado	Número(12,2)	Campos: (13) / (8)	Derivado

Tipo: Fact Table  
Nombre: Consumos / Prótesis\_Ortesis\_Optica

Nro	Campo	Tipo de Dato	Descripción
1	Clave_tiempo	Númérico(10)	-
2	Clave_Geografía	Númérico(10)	-
3	Clave_Afiliado	Númérico(10)	-
4	Clave_Diagnóstico	Númérico(10)	-
5	Clave_Plan	Númérico(10)	-
6	Clave_Responsable	Númérico(10)	-
7	Clave_Prótesis_ortesis_optica	Númérico(10)	-
8	Cantidad_prótesis_ortesis_optic	Númérico(10)	La cantidad de prótesis y óptica utilizados por los afiliados
9	Total_facturado	Númérico(12,2)	Monto facturado correspondiente a las prótesis y óptica
10	Total_liquidado	Númérico(12,2)	Monto liquidado correspondiente a las prótesis y óptica
11	Total_bonificado	Númérico(12,2)	Monto bonificado correspondiente a las prótesis y óptica
12	Total_debitado	Númérico(12,2)	Monto debitado correspondiente a las prótesis y óptica
13	Promedio_facturado	Númérico(12,2)	Monto promedio facturado correspondiente a las prótesis y óptica
14	Promedio_liquidado	Númérico(12,2)	Monto promedio liquidado correspondiente a las prótesis y óptica
15	Promedio_bonificado	Númérico(12,2)	Monto promedio bonificado correspondiente a las prótesis y óptica
16	Promedio_debitado	Númérico(12,2)	Monto promedio debitado correspondiente a las prótesis y óptica

Tipo: Fact Table  
Nombre: Consumos / Prácticas Médicas

Nro	Campo	Tipo de Dato	Descripción
1	Clave_tiempo	Númérico(10)	-
2	Clave_Geografía	Númérico(10)	-
3	Clave_Afiliado	Númérico(10)	-
4	Clave_Diagnóstico	Númérico(10)	-
5	Clave_Plan	Númérico(10)	-
6	Clave_Responsable	Númérico(10)	-
7	Clave_Práctica	Númérico(10)	-
8	Cantidad_consumos	Númérico(10)	La cantidad de practicas realizadas por los afiliados
9	Total_facturado	Númérico(12,2)	Monto facturado correspondiente a las prácticas médicas
10	Total_liquidado	Númérico(12,2)	Monto liquidado correspondiente a las prácticas médicas
11	Total_bonificado	Númérico(12,2)	Monto bonificado correspondiente a las prácticas médicas
12	Total_debitado	Númérico(12,2)	Monto debitado correspondiente a las prácticas médicas
13	Promedio_facturado	Númérico(12,2)	Monto promedio facturado correspondiente a las prácticas médicas
14	Promedio_liquidado	Númérico(12,2)	Monto promedio liquidado correspondiente a las prácticas médicas
15	Promedio_bonificado	Númérico(12,2)	Monto promedio bonificado correspondiente a las prácticas médicas
16	Promedio_debitado	Númérico(12,2)	Monto promedio debitado correspondiente a las prácticas médicas



---

## Referencias

---

Boh, Kathy. *Converting Data for Warehouses*.  
<http://www.dbmsmag.com/9706d15.html>.

Finkelstein, Richard. *Understanding the need for On-Line Analytical Servers*  
President, Performance Computing, Inc. ComShare.

Friend, David. *Una Introducción al Procesamiento Analítico On-Line (OLAP):  
Una explicación de la tecnología y terminología multidimensional*, Pilot Software,  
Inc. 1995.

Gupta, Vivek R. *An Introduction to Data Warehousing*. System Services  
Corporation, Chicago, 1997. <http://www.system-services.com>

Herb Edelstein and Janet Millenson. *How to succeed with End-User data access*.  
Euclid Associates.

Inmon, W.H.. *Building the Data Warehouse*, Second Edition, Wiley Computer  
Publishing, John Wiley & Sons, Inc. ,1996.

Jones, Greg. *Building Effective Data Marts*. Oracle Corporation, 1998.  
<http://www.oramg.com/oracle/98-May/data.html>

Kimball, Ralph. *The Data Warehouse Toolkit, Practical Techniques for building  
dimensional Data Warehouses*, John Wiley & Sons, Inc. ,1996.

Kimball, Ralph. *Meta Meta Data Data*. <http://www.dbmsmag.com/9803d05.html>

Kimball, Ralph. *Preparing for Data Mining*.  
<http://www.dbmsmag.com/9711d05.html>

Prabel, Frank. *OLAP for the masses: Delivering OLAP technology to the broader market*. Technical White Paper Series.

Orr, Ken. *Data Warehousing Technology*. Ken Orr Institute, 1996.  
<http://www.kenorrinst.com/dwpaper.html>

Sakaguchi, Toru, Frolik, Mark. *A review of the Data Warehousing Literature*. University of Memphis.

-----, *The Data Warehouse. The Competitive advantage for the 1990s*, A Red Brick Systems White Paper, June 1, 1995.

-----, *OLAP: Conversión de Datos corporativos en Inteligencia Empresarial*. White Paper I., 23/06/94.

-----, *What is a DataMart?*, <http://www.d2k.com/d2k/library2.htm>.

-----, *Data Warehousing: Empowering end users for enterprise-wide decision making*, Technical White Paper Series.

-----, *Decision Support ViewPoint. An enterprise-wide data delivery architecture*. White Paper prepared by MicroStrategy Incorporated.

-----, *Designing the Data Warehouse on Relational Databases* Stanford Technology Group, Inc.,.

-----, *An Arbor Software White Paper. Arbor Software, The Role of the Multidimensional Database in a Data Warehousing Solution*.

DONACION.....  
\$.....  
Fecha..... 3-10-05  
Inv. E..... Inv. B..... 2085

TES
99/13



BIBLIOTECA  
FAC. DE INFORMÁTICA  
U.N.L.P.



TES  
99/13  
DIF-02085  
SALA



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMATICA  
Biblioteca  
50 y 120 La Plata  
catologo.info.unip.edu.ar  
biblioteca@info.unip.edu.ar



DIF-02085